PERSONALNI
računari
75
SEPTEMBAR
1991.
CENA 100 D

SOFTVER

AUTODESK 3D STUDIO
MASM 6.0
RECOGNITA PLUS 1.2
TRUETYPE
JPI MODULA-2 2.0
MATHEMATICA 386 1.2
LOTUS ADD-IN TOOLKIT
BRIEF 3.0
QEDIT

JEZICI

FORTRAN 90

HARDVER

MLACOM 486
SX: NOVA KLASA AT NA 25 MHZ



DODATAK NA 48 STRANA

DOS 5.0

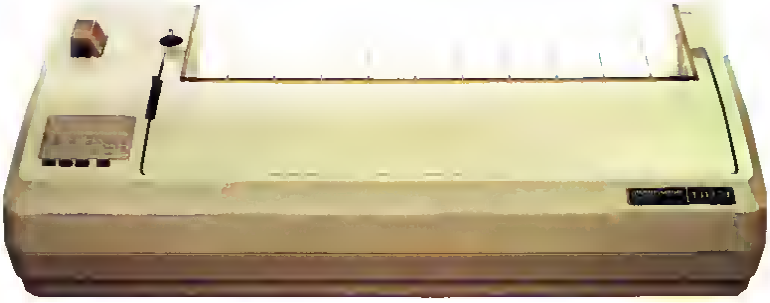
YU ISSN 0352-7271
9 770352 727009

IBIS SYS
ekonomski biro

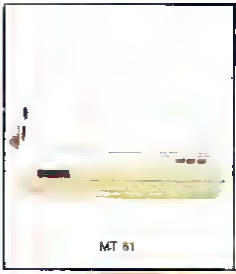
GENERALNI ZASTUPNIK ZA SFRJ

MANNESMANN
KIENZLE
MANNESMANN
TALLY

MT 131/24 KOLOR



- 24 IGLICE
- A3 FORMAT
- ŠTAMPA U BOJI
- BRZINA 250 z/s
- ČETVOROBOJNA TRAKA
- PARALELNI PORT SA 17 kB MEMORIJE
- 1 ORIGINAL I 3 KOPIJE
- 256 ZNAKOVA


MT 61


MT 340


MT 305


MT 660

IBIS SYS
SOFTWARE HARDWARE INŽENJERING

PROIZVODNJA SW I HW
ELEKTRONSKA OBRADA PODATAKA
BIROOPREMA I MALOPRODAJA
SERVIS
ZASTUPSTVO
KONSIGNACIJA

11071 Beograd
Pariske komune 22
Telefoni: 609 650, 674-831
Telex: 12384 IBIS YU
Žiro račun: 60615-601-50017
Telefax: 699-081

PERSONALNI
računari 75

časopis za informatiku i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
septembar 1991. cena 100 din

Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17

Generalni direktor
Ilija Rapajić

Direktor Novinskog sektora
Zoran Nikodijević

Glavni i odgovorni urednik
Jovan Regasek

Zamenik gl. i odg. urednika
Esad Jakupović

Tehnički urednik / sekretar redakcije
Natalija Desančić

Urednik za hardver
Dejan V. Vesalinović

Stručna redakcija
Vesna Ćosić (aktuelnosti) Nenad Baloćanin (baza podataka) Miodrag Lovrić (radne tabele) Dejan Ristanović (software), Jovan Skuljan (programiranje), Dejan Vesalinović (hardver), Dragutin Vulović (mreža) Zoran Životić (software)

Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III

Telefoni
653-748 (redakcija)
650-528 (prodaja)

Agencije BIGZ-a
(011) 651-793, 653-585

Telefaks
11455 BIGZ YU

Telefaks
(011) 648-140 647 965

Pretplata za zemlju
Za jednu godinu 1080
Za šest meseci 540
(na žiro-račun D.P. BIGZ, 60602-603-23254)

Pretplata za inostranstvo
Za jednu godinu 51 USD, 80 DEM, 67 CHS, 27 GBP 271 FRF
(na devizni račun D.P. BIGZ 60811 620-16101-620701-999-03377)
Zbog usporenog prenošenja uplata preko banke molimo pretplatnike da nam posle svake nove uplate odmah pošalju fotokopiju uplatnice

Rukopisi se ne vraćaju

SEZAM
sistem za modemske komunikacije
(011) 648-899 24 časa dnevno

Upravnik sistema
Jovan Regasek

Administratori sistema
Zoran Životić i Dejan Ristanović

Moderatori
Jovan Bulajić (datoteke), Aleksandar Damnjanović (PC SOFT) Slobodan Kalanć (ORKA) Vladimir Kostić (PC SOFT) Vladimir Kristonjić (VICE-VA) Esad Jakupović (RAČUNARI) Dalibor Lanik (ATARI ST) Dejan Ristanović (NOVOSTI I KOMUNIKACIJE) Bojan Tepavčević (FORUM) Dejan Vesalinović (PC HARD) i Zoran Životić (SEZAM)

Održavanje sistema
Natalija Desančić

Softver
Zoran Životić (glavna aplikacija) i Dejan Ristanović (eksterni programi)

Hardver
Server: COMPAQ DESKPRO 386/25, 3 MB RAM-a tvrdi disk 300 MB (COMPAQ, Minhen) Mreža: dve stanice COMPAQ 286N/12 1 MB RAM-a (COMPAQ Minhen) tri stanice „Ultra" AT 12 MHz, D-LINK DE 100 LanSmart („Ultra" Skopje) Modemi: VIVa (Telekomunikacioni centar, Beograd)

SADRŽAJ

4 Novosti
ŠTA IMA NOVO
6 Tržište
ŠTA ĆE BITI
8 Računari / Mlecom 488
TERMINATOR II
10 Računari
SX: NOVA KLASA
12 Računari
PROCESOR KOGA TREBA UBITI
14 Ploteri / CAMM-1
REZAČNICA SLOVA
15 Grafički softver / 3D Studio
ŽIVE SLIKE
17 Grafičke kertice / NthEngine 150
BRZINA I PEDIGRE
18 Grafički softver / IconAnimete
IGRE SA SLIKOM
19 Asembleri / MASM 6.0
KRALJ U NOVOM U RUHU
22 Prepoznavenje teksta / Recognita 1.2
VEŠTINA ČITANJA
26 Digitalna tipografija / TrueType
TYPE I ILI TRUETYPE
30 Kompajleri / JPI Modula-2 2.0
VIŠE OD EGZOTIKE
34 Programski jezici / Fortran 90
USVOJENO KOMISIJSKI
36 Metematički softver / Mathematica 386 1.2
KRUNA ZA KRALJICU NAUKA
38 Softverske alatke / Lotus Add-In Tcolkit
LOTUSOVE ALATKE
40 Editon / Bnef 3.1
JOŠ UVEK NA VRHU
42 Editori / Qedit
PROBAJTE SLOBODNO
44 CAD / Definisanje krivih i površi
GEOMETRIJSKO MODELIRANJE
47 Operativni sistemi / Mekintoš 7.0
LOKALIZOVANI MEKINTOŠ
48 Algoritmi / Metaalgoritmika
UZBRANO SA DRVETA
50 Algontmi / Genetski algoritmi
UČIMO OD PRIRODE
52 Simulacije / TurboPascal: Life3D
ŽIVOT JE LEP
54 Tehnike programiranje / Zaštite softvera
POTPIŠI SE I POČNI
56 Tehnike programiranje / Windows
PUCAMJ KROZ PROZOR
58 Tehnike programiranja / Clipper
PREVIEW U KLIPERU
62 Programski jezici / C++
DINAMIČKI OBJEKTI
67 Dejan Ristanovič
BAJTOVI LIČNE PRORODE
68 Dejanove pitalice
NAJMANJI OD NAJVEĆIN
70 Zoran Životić
MOJA ŠKOLA C-a
72 Komunikacije
SEZAM BILTEN



Evo ponove na naslovnoj strani naše već "idealne" priče o lepotici i zveri. Kao što se vidi po njenom osmehu, lepotica "na turne", Mara Radulović iz Beograda, nije se ni najmanje uplašila ove velike zveri – moćnog računara firme "Mlecom" koji u klasi 486 na 25 MHz predstavlja vrhunac. Ili je našu lepoticu ohrabrilo prisustvo male ekipe – Tamara Ćuk (styling), Darla Milošević (make up) i Nenad Marjanovića (fotografije) – ili ona zaprave ume sa dobrim mašinama...

oktobarski broj
časopisa
računari
izlazi iz štampe
15. oktobra 1991.

Na osnovu Mišljenja Sekretarijata za informacije SR Srbije broj 413-01-48 od 4. 2. 1991. na ovaj časopis plaća se osnovni porez na promet po povlašćenoj tarifi od 3 odsto.

Tržište

ASHTON-TATE ODLAZI

Mnogi elementi najavljene fuzije Borlanda i Ashton-Tate-a još uvek nisu dovršeni, ali su neki detalji ipak procureli u javnost. Jedna od glavnih novosti je nestanak imena Ashton-Tate, nova kompanija će se jednostavno zvati Borland i Borland i A-T su u protekloj fiskalnoj godini imali prihod od približno 230 miliona dolara sa 1000, odnosno 1700 zaposlenih, dok su Lotus i Microsoft u istom periodu imali prihod od približno 893 miliona i 1 6 milijardi dolara respektivno

Neki A-T-ovi proizvodi se, po svemu sudeći, neće uklopiti u Borlandov razvojni plan, naročito *Macintosh Full* linija (*Full Write, Full Impact*) i *Control Room utility* Borland je, sa svoje strane, već odbacio celokupnu svoju Macintosh liniju Ni u Borland ni A-T još uvek ne komentarišu buduće planove u vezi ove linije Predstavnik Borlanda je, doduše, izjavio da se radi na *Windows* orijentisanom tekst-procesoru, što znači da Borland namerava da još jednom proširi svoju gamu proizvoda Najasna je i situacija oko dBASE-a, budući da A-T razvija *dBASE for Windows*, a Borland radi na *Object dBASE*, ali predstavnici obe kompanije tvrde da će *dBASE* preživeti, kao i Borlandovi *Paradox, Quattro* i ostali paketi, uključujući i *Interbase*, paket namenjen robusnijim okruženjima od samog DOS-a (32-bitni OS/2, UNIX VMS)

COMPAQ SNIŽAVA CENE ZA LTE

Nova runda snižavanja cena nije zaobišla ni Compaq – notebook računari LTE serija sada nose etikete sa sledećim cenama LTE/286 sa 20-MB hard diskom košta $1849 (preme ranijih $2499), isti model sa hard diskom od 40 MB sada košta $2249 prema ranijih $2799, dok za cenovni LTE model baziran na 80C86 procesoru sa 20-MB hard diskom treba izbrojati $250 manje, tj. $1749 Za ove cene je rečeno da mogu da variraju od dilera do dilera

Štampači

SEIKO POSTSCRIPT U KOLORU

Osnovna odlika novog *ColorPoint PSX* kolor štampača firma Seiko je veliki bafer koji može da primi celu slliku i tako oslobodi računar odmah po završenom slanju Reč je o PostScript kompatibilnom kolor štampaču rezolucija 300 dpi namenjenog PC-kompetibilcime, Macintoshima i UNIX radnim stanicama sa mogućnošću simultanog pristupa Štampač koristi tahnologiju termalnog transfera koju kontroliše 25-MHz Intel 80960 RiSC procesor, a raspolaže i HPGL emulacijom

Na raspolaganju su dva modela Model 4 sa maksimalnim formatom papira od 8 9 x 11 inča (A4) koji raspolaže sa 10 MB RAM-a, i Model 14, tzv *tabloid model*, sa maksimalnim formatom od 11 x 17 inča (A3) i 18 MB RAM-a Oba modela su standardno opremljana AppleTalk paralelnim i serijskim portom, a tu je i kolekcija od 39 Apple LaserWriter II kompatibilnih fontova *ColorPoint PSX* pruža najveću fleksibilnost standardnog papira – najmanje moguća margina iznosi 1/8 inča (približno 4 2 mm) Porad toga, profesionalnim grafičkim umstnicima je pružena mogućnost korišćenja predimenzioniranog papira tako da se otisak „razlije" do same granice standardnih dimenzija (super-size technology)

Seiko ColorPoint PSX se nalazi u prodaji po cenama od $9499 za Model 4 i $12 999. Memorija oba modela se može proširiti do 34 MB

Kontakt adresa Seiko Instruments USA Inc., Graphics Devices and Systems Division, 1130 Ringwood Court, San Jose, CA 95131,(408) 922-5800; fax, (408) 922-5840

NOVI FUJITSU 24-PINAC

Fujitsu America je pustio u prodaju novi 24-igličnl štampač koji treba da bude direktan konkurent novim Epson, Panasonic, Okidata i NEC modelima. *Fujitsu DL 3600*, kako glasi puni naziv novog modela, je opremljen softverom koji se na leporučuje ni uz jedan drugi štampač na tržištu *ZipType*, TSR program omogućava upotrebu štampača kao pisaće mašine, kontrolu tastatura i programiranje maski za najčešće korišćena obrasce *DLMenu* služi za konfigurisanje štampača nezavisno od kontrolnog panela a *Creative Faces* služi za rad sa *PostScript* promanljivim fontovima koja treba download-ovati sa računara, a raspoloža i sa kolor opcijom uz upotrebu četvoroodjna trake koja apada u standardnu opremu.

Fujikeu tvrdi da je DL 3600 brži od bilo koja mašine u toj kategoriji – 360 cps u draftu (poređenja radi, Epson LQ 1170 raspolaže brzinom od „samo" 330 cps, takođe u draftu). Cena $849

Kontakt adresa: Fujitsu America, Computer Products Group, 3055 Orchard Dr, San Jose CA 95134, 800 626-4686, 408 432-1300

Operativni sistemi

PENDOS ZA PEN PC RAČUNARE

Za sada još nema mnogo komercijalnih programa namenjenih računarima koji umesto tastatura koriste svetlosnu olovku *Communication intelligence Corp.* se nada da će uspeti da popravi ovakvo stanja uz pomoć svog *PenDOS*-a koji treba da omogući DOS aplikacijama da rade na Pen PC-ima *PenDOS* se sastoji iz dva dela prvi deo je program sa prepoznavanje rukopisa za koji firma tvrdi da je u stanju da raspozne eve alfanumeričke karaktera, a drugi deo pakete pradstavlja ulazni interfeja koji radi iznad DOS-a, rasuma gestovna komanda i emulira miše

Kombinacija *PenDOS*-a i *MSWord*-a bi u prekel omogućila korisniku da selektuje tekst pokratom olovka naprevi promanu odredenim znakom (recimo X za brisenje), ili da unosi tekst pisanjem unutar prozora. *PenDOS* ipak na radi se svim kombinacijama DOS aplikacija i sistema, budući da zahteve 330K *extended* memorija za modul zadužen za prapoznavanje rukopisa i 50K DOS mamorija za *PenDOS* interfeja Paket radi bez problema uz *Lotus* 1-2-3, *MS Word, Quattro Pro, Harvard Graphics* i *Norton Commander PenDOS* se na tržištu očekuje u septembru, a cena još nije poznata

Kontakt adresa: CIC, 275 Shoreline Dr, Redwood Shores CA 94065; 415328-1311; fax 415 325-8681

Mreže

UBRZIVAČ ZA ETHERNET

The EtherSwitch EPS-1500 je naziv uradeja koji je u stanju da po rečima pradstavnika kompanije Kaplana Inc, poveća brzinu prenosa sa standardnih Ethernetovih 10 na punih 70 miliona bitova u sakundi. Princip rada se zasniva na uspostavljanju višestruke paralelne veze izmedu LAN-ova i hostova pomoću brzih prekidače Slatem podržava tanke, debele, uvrnute i fiber-optičke kaolova EPS-1500 se može kupiti u više konfiguracija: uredaj sa 8 portova košta $12 995, za 15 portova treba platiti $19 995, a cena svakog dodatnog porta iznosi $1000

Kontakt adresa Kaipana, 125 Nicholson Lane, San Jose CA 95134, 408428-1150, fax 408 428-1161

Komercijalni softver

BRIEF DOBIJA PLUSEVE

Dodatak za poznati editor Brief. *Briefor C++*, je napravljen se ciljem prilagođavanje za C++ pomoću specijalnog browser-a Na taj način se korisniku omogućave da pretražuje klase i struktura u pop-up prozoru i to po hijerarhiji klase, varijablama ili funkcijama. Moguće je istovremeno skeniranje više fajlova uključujući i fajlove na dsku, na samo ona u memoriji. *Briefor C++* automatski učitava relevantne fajlove i pozicionira kursor na traženi siement. Cena $129 za *Briefor C++*, ili $339 u paketu sa Brief 3.1

Kontakt adresa Solution Systems, 372 Washington St., Wellesley, MA 02181, (500) 677-0001 or (617) 431-2313, fax, (617) 740-0089.

IMAGEMAN

ImageMan je objektno orijentisana *Windows* biblioteka koja daje *Windows* aplikacijama mogućnost prikazivanja i štampanja velikog broja formata istim setom funkijskih poziva *ImageMan* podržava TIFF PCX, ankapsulirani PostScript, *Windows* Metafile i bit-mapirane slike, snaodaven je *Dynamic Link* bibliotekom i radi sa bilo kojim programskim jazikom koji podržava DLL poziva, kao što su C++, *Turbo Pascal* for *Windows Visual Basic, Smalltalk/V, Aotor, Realizar* i drugi Cena $395, sa kompletnim izvornim kodom $995

Kontakt adresa: Data Techniques, Inc., 1000 Business Center Dr., Suite120, Savannah, GA 31405, (800) 999-8003 or (912) 651-8003.

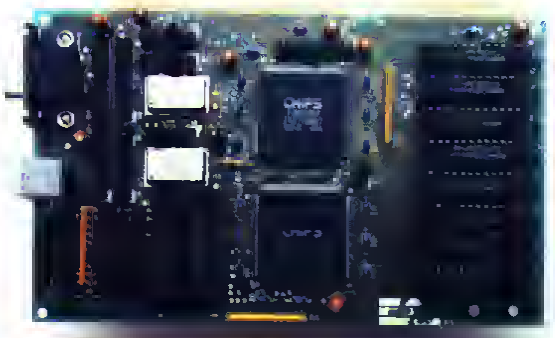
EZ-INSTALL 4.0

Firma *Software Factory* radi na varziji 4 0 programa *EZ-Install* koji služi za pojadnostavljivanje instalacionog procesa sa komarcijalne programa koji rade pod *Windows*. Nove *Windows* aplikacija naima često zahtevaju od korisnika da postave specifične parametre u INI fajlove da bi radili kako treba. Reč o osetljivom procesu koji treba vrlo pažljivo izvršiti – u protivnom dolazi do onesposobljavanja celog *Windows* okruženja *EZ-Install 4 0* rešava ovaj problem tako što obavlja ceo posao automatski – od korisnika se očekuje samo da naznači sekcija i ključna imena koja treba provariti modifikovati, zameniti ili ubaciti u odgovarajući iNI fajl Ostala karakteristika oduhvatalju grupnu instalaciju sa automatizovanim menjima, dinamičku promenu bež fajlova, izmanjenu i dopunjenu EZARC kompresionu rutinu i neograničena mogućnosti kreiranja direktorijuma EZ-Install radi kao DOS aplikacija, a moguće je dobiti i OS/2 varzlju Cena: do 1. novembra iznosi $199, dok će posle toga iznositi $249 OS/2 varzija će koštati $349, a varzija koja radi pod oba operativna sistema će koštati $449

Software Factory, Inc., 17610 Midway Rd., Suite 134-222, Dallas, TX 75287, (214) 490-0835, fax, (214) 306-4552

Video kartice

SUPER BRZA GRAFIKA



Pored sada već uobičajenih SuperVGA čipova, mnoge kompanija koja se bave VLSI „biznisom" počele su da zasipaju tržišta specijalizovanim grafičkim čipovima namenjenim ubrzanju rada u pojedinim aplikacijama, naročito onim zahtevnim kao što su *AutoCAD* ili *Windows*. Poole Weltekovih 5086 i 5186 ubrzivača *Windowsa* i Chips & Technologies, sada već gigant VLSI industrije, predstavio je svoje pulene. Prvi je 82C480, grafički koprocesor (pazite, na grafički procesor!) za 386 i 486 koji je 100% registarski kompatibilan se IBM 8514A 82C480, inače ugrađen na NthEngine/150 koju smo testirali, znatno je brži od IBM 8514A i WD PWGA1 koprocesora. 82C480 je, uz to kompatibilan, i sa novim IBM XGA, ali i brži od njega, a omogućava rezolucije do 1280x1024 bez preplitanja.

Druga novost je PUMA. PUMA (Programmable Universal Micro Accelerator) je naziv za dva čipa koja zajedno neposredno ubrzavaju grafički intenzivne operacije. Prvi je PUMA-P procesor 84C2001, koji radi na 50 MHz. Koristeći 64-bitni mikrokod, PUMA-P može izvršiti više grafičkih operacija u jednoj procesorskoj instrukciji koja će uzeti između dva i pet takt ciklusa. Procesor je u potpunosti programabilan (vi sami ćeštinišete naredba mikrokoda koje procesor čita preko posebne ultrabrze mikrokod magistrale) tako da, pored strahovitog ubrzanja ekranskih operacije, ubrzave i red sa laserskim printerom, na primer. Prkom procesor na zahteve nikakav VRAM, nego samo lokalnu statičku memoriju se mikrokod i opcionu memoriju se podatke. To omogućava drugi čip, PUMA-I. On se direktno vezuje na PUMA-P s jedne, i ISA ili EISA magistralu s druga strana. PUMA-I, ili 84C2002, omogućava PUMA paru da bude „bus master" na sistemskoj sabirnici i pristupa svim resursima sistema, od RAM do grafičke kartice.

Ako hoćemo da izbegnamo eventualne usporenja sistemska magistrale, PUMA može biti integrisane na VGA, 8514A ili XGA karticu i, porad ostalog, biti i vrlo brz „display list" procesor – višestruko brži i od TI 34020 grafičkog procesora. Pritom je sva dodatna logika koju PUMA traži svedena na par TTL kola, što će cenu celokupne otveri održati veome niskom – možda dvostrukom cenom sadašnjih Tseng ET4000 kartica. Uz novi 8514A ili XGA (VGA kompatibilan) kontroler i kolo se neposredan ulaz video signala, što vam divnog grafičkog podsistema ze male pare. A pošto je IBM odlučio da potpuno otvori XGA specifikaciju i ueput se učlanio u VESA udruženja (znači more novih superbrzih klonove ja ne vidiku), stveri postaju jeanije.

Nebojša Novaković

JOŠ VEĆA KOMPRESIJA

Optilink/Compresa 3.0 je program koji vrši jednostepeno komprimovanje i tako omogućava znatnu uštedu prostore na diaku. Prama rečima predstavnika kompanije, uključeni linker radi četiri do deset puta brža od MS Link-a i omogućave obrazovanje samorsepakujućih izvršnih fajlova (pri izvršavanju programe ugrađena dekompresiona rutine automatski vraća program u originalni oblik). Glavne osobina nova verzije su raspologanja se čak 6132 meeka (overlay). Indirektni pozivi, EMS/XMS kaširanje, auto-alokacije biblioteka, itd. Pored verzije 3.0, ista kompanije nudi i Optilink/2 namanjen 16-bitnom OS/2 i *Presentation Manager* programime i Optilink/*Windowa*. Cana je $350 po programu.

Kontakt adresa: SLR Systems, 1622 N. Main St., Butler, PA 16001, (412)282-0864, fax, (412) 282-7965.

Ekspanzione kartice

NOVA SX KARTICA ZA 286

Nove ekspanzione kartice firme *Cumulus* omogućava korisnicime „podsenja" 286 računara u 386 klasu. Ovo nije prva takve kartice, ali novost je u tome što se, umesto praznog slota, postavije u prsena 286 podnožje 286 mašine se 386SX procesorom nije isto što i pravi 386SX sistem – brzina internog 286 klocka limitire maksimalno moguću brzinu – ali ipak daju kompatibilnost se 386-bazirarnim softverom kao što su *Windows* i većina CAD programa.

Kartica radi na 15 MHz i podržava 387SX matematički koprocesor. CAD programi na 286 sistemu se 386SX karticom i 387SX koprocesorom rade do pet puta brže nego na klasičnoj 286/287 kombinaciji. Kartica uz koju se dobija i MS *Windows*, se proizvodi u pet različitih modale saviano od tipa računara, e uz karticu treba dokupiti i set kablove ($25) se rad se 387Sx koprocesorom.

Kontakt adresa: Cumulus Corp., 23500 Mercantile Rd, Cleveland, OH 44122, 216 464-2211, fax 216 464-1150

NOVI HARVARD GRAPHICS

Dobro poznati program Harvard Graphics je u junu dobio verziju 3.0. Glavna razlika se ogleda u tome što program automatski određuja najpogodniji dizajn se svaki grafik. Za to se koristi koncept elegalice koja se sastoji od elemanata kao što su naslovi, podaci, fusnote i legende. Menjanjem atributa bilo kog od ovih elemenata, menja se i kompletan izgled grafa. Porad toge, nove opcija su i automatski prenos reči u novi red (Word Wrap), osnovna animacija i automatsko reformatiranje. Ovim nije iscrpljen spisak novih opcija, e od onih koje nisu pomenute, pažnju privlači pregled cele tabele (spreadaheet view) koja je naprevljene tako da omogućave lako ubacivanje podataka iz tabele u dijagram – umesto navođenja brojeva pojedinačnih ćelija, sada se cela tabele prikazuje na ekranu a korisnik samo treba da vizuelno pokaža ćelije koja žali. Paket se isporučuje se 15 fontova promanjive veličina u formi *Bitstream Speedo* fontova i 35 crtačkih alata. Nove verzije košta $595, dok registrovani korisnici prethodne verzije treba de doplata $125.

Kontakt adresa: Software Publishing Corp., 1901 Landinga Dr, PO Box 7210,Mountain View, CA 94039-7210, 415-962-8910

LOTUS I DALJE PODRŽAVA XT

U Lotusu i na pomišljaju da napuste podršku XT-klase računare umesto toga razvijaju nove verzija svog popularnog paketa 1-2-3 v 2. Najnovije verzije nosi oznaku 2.3 i pored određenih kozmetičkih izmena sada raspolaže i *Bitstream Speedo* fontovima se generisanje teksta i nsalove, novim alatima se kreiranje grafova i dijagrame, kao i paletom od 224 boja, tekat editorom koji automatski uređuja tekst oko grafa i alatima za slobodoručno crtanje. Uporedo se novim pretežno grafičkim pristupom poboljšana je i podrška miša – sada je dovoljno kliknuti na bilo koje polje de bi se omogućila operacije kao što su dodavanje, manjenje ili brisenja informacija ili izvršavanja komandi. Nova verzije 2.3 raspolaže i funkcijema sa izdvajenje svih podataka povezanih se odradenom ćelijom.

Lotus 1-2-3 V 2.3 može de radi već na XT-u se 384 K RAM-a, ali de bi bile dostupne WYSIWYG funkcije, potrebno je najmanje 512 K. Lotus očekuje da će mnogi korisnici ostati pri cenovnim DOS mašinama bar još sledeće podina. tako de se radi i na novim poboljšanjima verzija 2. Cene verzija 2.3 je $495, a Network Server i Noda Editiona se očekuju u najskorije vreme po ceni od $755 i $465 respektivno.

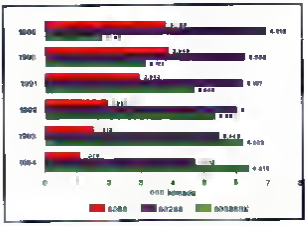
Kontekt adresa: Lotus Development Corp., 55 Cambridge Parkway, Cambridge, MA02142, (617) 577-8500

ŠTA ĆE BITI

Šta se događa na svetskoj pijaci? Šta nas čeka u narednom razdoblju, kako do kraja godine, tako i u narednim godinama? I šta se događa koda nas, u ovim ne baš veselim vremenima? Zavirili smo ovih dana malo u našu staklenu kuglu. I evo šta smo videli.

Dejan V. Veselinović

U svetu koji ne haje mnogo za naše lokalne sportove tehnika i tehnologija se uporno razvijaju dalje. Iako računarska industrija u svetskim i posebno američkim okvirima prolazi kroz period recesije, ipak se radi o blažem tržišnom hiru pre nego o ozbiljnijoj stagnaciji. Nove tehnologije omogućavaju dalje pojeftinjenje svih proizvoda, mada proizvodi koji konstantno menjaju mehaniku (tvrdi diskovi, disketne jedinice) daleko sporije opadaju po ceni nego čista elektronika.

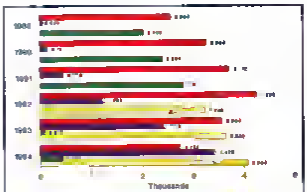


Slika 1. Stanje i razvoj tri glavne procesora firme Intel

U SVETU...

Slika 1 prikazuje stanje i razvoj tri osnovne kategorije proizvoda. Matične ploče sa procesorom 8088 će do 1994. godine pasti na manje od trećine broja proizvedenih komada na vrhuncu svoje moći, 1990. godine. Ovaj podatak ne treba da vas čudi – iako Evropa i Amerika odustaju od ove tehnologije, ima dosta delova sveta gde je ona zbog svoje cene još uvek veoma značajne (Afrika i Azija).

Isti opšti trend važi i za ploče sa procesorom 80286, koji je svoj vrhunac proizvodnje doživeo čak godinu dana pre starijeg i slabijeg brata, što je svojevrsan apsurd, mada se ne radi o mnogo manje dvostruko većem apsolutnom broju. Za razliku od ovih, broj ploča se procesorom SX će porasti za 3.61 puta u periodu 1989-1994. i u polupunosti preuzeti primat na ekonomskom delu tržišta.



Slika 2. Trendovi rasta Intel i Motorola procesora

Slika 2 prikazuje trendove rasta broja proizvedenih matičnih ploča sa procesorima firme Intel (svi su 32-bitni) i porodicom procesora firme Motorola (porodica 88XXX, svi članovi zajedno). Interesantno je primetiti da se predviđa prilično veliki porast prodaje Motorolinih procesora, za koji se očekuje udvostručenje u petogodišnjem periodu 1989-1994. A zašto da ne, kada se radi o veoma dobrim procesorima, koji predstavljaju jaku alternativu Intelovim čipovima?

Iako će prema ovim nepotpunim prognozama Motorola popraviti svoju poziciju u odnosu na Intel (slika 3), ipak treba napomenuti dve stvari: prvo, sve prognoze se odnose na tajvansko tržište, čime su izostavljeni hongkonški, singapurski, japanski i američki proizvođači, koji mahom proizvode modele sa procesorima Intela, i drugo, iako će se odnos popraviti sa prvobitnih 7.5 : 1 u korist Intela, ipak će ostati na 4.5 : 1 u korist Intela. Najzad, opaska opštijeg tipa je da se radi o prognozama, koje mogu ali ne moraju biti tačne.

Slika 4 prikazuje prosečne godišnje stope rasta proizvodnje matičnih ploča za petogodišnji period. Primetićete da uprkos veoma velikim apsolutnim brojkama, ploče na bazi procesora SX zapravo imaju sasvim razumne stope rasta od 19.54%. Slično ovome, ploče na bazi procesora Motorola takođe pokazuju razumnu stopu rasta od 14,15% godišnje, mada strogo ekonomski posmatrano, oba brojke su veoma dobre i ukazuju na veoma dobar rast i razvoj tržišta.

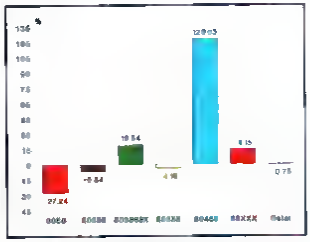


Slika 3. Tržišno učešće procesora Intel i Motorole u apsolutnim brojkama

Zato je slopa rasta proizvodnje ploče se procesorom 80486 gotovo neverovatna: apsolutni sam svakog proizvođača prosečno 123.03% godišnje! Blago Intelu, blago nama, jer bi valjda neki deo tog broja mogao da zaluta i do nas. Nadajmo se.

Koliko Intel i Motorola dobro stoje, toliko se ne može pisati drugima, koji će prema prognozama zabeležiti negativnu prosečnu godišnju stopu rasta od 0,75%. To je valjda po onoj narodnoj „Dok se jednome ne smrkne, drugome ne sviće". Verovatno je da će se u ovom segmentu stvari odvijati tako što će neki potpuno nestati, a drugi ipak rasti, ali sa veoma umerenim godišnjim stopama.

Fascinira to da će prema ovim prognozama samo Tajvan proizvesti gotovo 23 miliona matičnih ploča 1994. godine. To znači da će se svetska produkcija, što obuhvata i druge proizvođače, kretati na nivou 40-45 miliona računara godišnje, prema nekim prognozama, sada u svetu ima oko 60 miliona IBM i kompatibilnih računara. Kada ove brojke primenimo na sadašnje stanje i tako pretpostavimo da će trenutna godišnja proizvodnje predstavljati zamenu za uništene, polomljene i na druge načine upokojene ploče, proizlazi da će 1994. godine u svetu biti oko 160 miliona računara, odnosno bar tri puta više nego danas.



Slika 4. Prosečni godišnji trendovi razvoja tržišta po procesorima, 1989-1994.

...I KOD NAS

Mi smo uvek nešto ekstra, dok se svet naoružava računarima, mi se naoružavamo računarima tipa „kalašnjikov", laptop model AK-47 svaki sa bar po jednom rezervnom baterijom. Zbog toga ni prognoze nam nisu ružičaste.

Slika 5 prikazuje projekcije prodaje zaključno sa ovom godinom. Pošto kod nas nema pouzdanih podataka, sve se svodi na puku prognozu, pa ove brojke nemojte uzeti baš doslovno. Dobijene su blic ispitivanjem tržišta u Ljubljani, Zagrebu i Beogradu. Konačne brojke su ekstrapolirane iz brojki o mesečnoj dinamici prodaje, koja je prikazana na slici 6.

Zbog poznatih događaja, a i šire, broj prodatih računara je trebao da bude osetno manji no prošle godine, kada problema sa nabavkom deviza i plaćanjima ka inostranstvu nije bilo bar ne u prvih 10 meseci godine. Šta se posle desilo, to svi znamo. Posledice toga se jasno vide u brojkama prodaje u prva tri meseca ove godine.

U četvrtom i petom mesecu dolazi do prilično naglog porasta, pre svega u Sloveniji, valjda je proradila „Radio Mileva" pa su ljudi i preduzeća požurili da se opreme na vreme, što nije trebalo. Delimično, ista stvar važi i za Hrvatsku. U junu mesecu prodaja ide dobro sve do 25. juna, kada u Sloveniji pre svega, a delimično i Hrvatskoj maltene propada. U julu mesecu, dolazimo do vrhunca prodaje za ovu godinu, i to najvećim delom u Srbiji, zapravo ponavlja se priča iz Slovenije u maju mesecu. Ljudi i preduzeća kupuju kao da je besplatno.

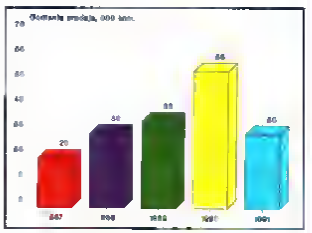
u šla smo se i lično uverili u više beogradskih preduzeća stoje ljudi lepo u redu parkirali kola plate i nose tovare li tovare Zaista nevero vetno ali ne bez osnova logika je bila „dat da pare veoma nesigurne u mutnim vremenima pratvonim u opremu koja će mi ionako trebati" Ovo je zapravo tržišni aspekal psihoze nesi gurnosti u ratnom stanju



Slika 5 Godišnje prodaje računera u Jugoslaviji, 1987-1991. (projekcije)

Naravno ovde se radi o efektu forsirane tražnje odnosno tražnje koje je sa perioda cele godine naglo sabijena u jedan mesec dana, a delimično i od novoformirane tražnje Nju su stvorili ljudi koji u normalnim okolnostima možda i ne bi kupili računar ili ne toliko komada U svakom slučaju ovakav tempo prodaje neće opstati do kraja godine, ako je tražnja nenormalno porasle tokom jula i ako ne dođe do velikih pogoršanje opšte situacije u zemlji tražnje u neradnim mesecima će jednostavno pasti ispod očekivanih nivoa

Sve u svemu dok prodaje računera u svetu raste kod nas opada dok se u svetu relativno stabilizuje, kod nas postaje sve podložnija političkim trenucima, situacijama od danas do sutra Umesto dugoročnog trenda rasta tražnje mi imamo ne relativnu već apsolutnu stagnaciju jer umesto rastuće ili bar relativno stagnirajuće vrednosti domaće valute imamo stalni pad i konačno vraćamo se u dane gelopiraju će hiperinflacije



Slika 6 Dinamika prodaje računara u Jugoslaviji po mesecima za 1991. godinu (projekcije)

Dođe čoveku da se u tuzi nasmeje naknvi šešir i zapeva „Ma dobro da uopšte još kupujemo računara i to je veliki uspeh" Od lepa želje da ove godine prebacimo brojku od 220 000 računara u zemlji nema ništa daj bože da dostignemo bar 200 000 a ni to nije mnogo verovatno Po ko zna koji put znanje je gurnuto ustranu zarad nekih viših (nižih?) interesa nije nam to prvi put ali pitamo se dokle ćemo tako koliko još puta možemo sebi to da dozvolimo?

STA IMA NOVO

Komunikacije

BRZI V.32 MODEMI

Kad su svojevremeno cene 2400 bps modema opale ovi su postali sasvim uobičajeni, standardni deo opreme sada ne istu stazu stupaju i V 32 (9600 bps) modemi – novi *Twincom 96/42* modem se brzinom od 9600 bps i cenom od svega $299 predstavlje prvu lastu Reč je o Heyes-kompatibilnom internom modemu koji raspolaže V 32 prenosnom protokolu koji obezbeđuje brzinu prenosa od 9600 bps i V 42 (MNP 2-4) i V 42bis (MNP 5) protokolima za korekciju greške i kompresiju podataka. Sa V 42bis protokolom, ovaj modem će biti sposoban da ostvari stepen kompresije od 4:1 odnosno efektivnu brzinu prenosa od 38 400 bps

Kontakt adresa Image Communications, 6 Caesar Place, Moonachie NJ 07074 (201) 935-8880

32-BITNI BEJZIK ZA 386/486

Firma *32 Bit Software* je ponudile bejzik namenjen isključivo 32 bitnim personalcima sa procesonma 386 i 486 pod nazivom *ZBasic-PC 4.6* Paket omogućava pisanje programe i na nižim mašinama uz uslov da se rekompajliraju u 32-bitnu verziju na odgovarajućem hardveru Osnovne osobine su grafika nezavisna od izlaznog uređaja, VGA i MCGA podrška, prepoznavanje svih grafičkih monitora istog tipa sofisticirana komunikacija sa interapt-baferom od 64 K koji podržava četiri porta istovremeno pn brzinama do 115 200 bps, podrška dva različita monitora u isto vreme, kompajliranje unutar editora i podrška mreže Sve to za $99 95

Kontakt adresa 32 Bit Software, 3232 McKinney, Suite 865 Dallas TX 75204, (800) 322-4879 or (214) 720-2051, fax (214) 855-0677

PROGRAM ZA PISANJE PROGRAMA

Cause, generator aplikacija čiji je prvobitni distributer bila firma Maxem sada dobija drugu šansu, ovog puta pod okriljem firme Cyma Reč je da podsetimo o generatoru aplikacija za Macintoshe i PC kompatibilne koji omogućava pisanje unakrsno-kompatibilnih programa nezavisno od sintakse i kodiranja Verzija 2 0 omogućeve unošenje poziva za programe napisane u nekom drugom jeziku i posredovanje podataka medu različitim Cause aplikacijama Dok je prethodna verzija radila samo sa mišem verzija 2 0 dozvoljava i upotrebu tastatura U planu je i *NetEffect* runtime modul za AppleShara NetWare i NetBIOS kompatibilne mreže Svaki paket ponaosob košta $495 a komplet $795

Kontakt adresa Cyma 1400 East Southern Ave Tempe AZ 85282, (602) 831 2607 fax, (602) 345-5703 At COMDEX Booth 206

LOTUS I ADOBE TYPE MANAGER

Lotus će ubuduće uz svoje nove *Windows* aplikacije uključujući i *1-2-3 for Windows* i *Freelance Graphics for Windows*, davati i *Adobe Type Manager* i 13 Adobe fontove Ova font tehnologije koja generiše jasne i precizne fontove promenljive veličine će takođe ići i uz buduće verzije pakete *Ami Pro* i *Notes*. Lotus i Adobe su potpisali licencni sporazum kojim Lotus dobija pravo na upotrebu Adobe fontove i tehnologije u svojim aplikacijama ATM se konsti u IBM verzijama OS/2 i, naravno, na Macintoshu

Windows scena

NOVI ČIP UBRZAVA WINDOWS

Kompanije *S3 Inc.* je predstavila novi čip projektovan specijalno za ubrzavanje grafike pod MS *Windows* i ostalim grafičkim korisničkim interfejsima kao što su OS/2 Presentation Manager i XWindow 86C911, kako glasi zvanična oznaka novog čipa je predviđen za ugradnju na osnovne ploče i grafičke kartice, ubrzanje postiže hardverskim preuzimanjem mnogih osnovnih grafičkih funkcija koje inače izvršava sam procesor Rezultat ovog pristupa je pored brzog obnavljanje ekrana, opšte poboljšanje performansi celog sistema, budući da glavni procesor tako troši mnogo manje vremena na grafičke posiove. Čip podržave sei izvpnmitivnih grafičkih funkcije potrebnih za glavne funkcije grafičkih konsnićkih interfejsa kao što su BitBlts (pomeranje i menjanje veličine prozora, pop-up menje i kontrolnih panela), transfer slika, popunjavanje površina, rasterska operacije, fontovi, crtanje linija, rad sa bojama i kontrola kursora.

Da bi se kontroler ne bezi 86C911 čipa mogao konstiti, potreban je specijalni grafički drajver – proizvođač je uz čip razvio i drajvere za poznatije grafičke interfejse i za neke DOS aplikacije kao što su AutoCAD, VersaCAD, Lotus 1-2-3,WordPerfect, Word, i Ventura Publisher Čip je kompatibilan na nivou registara sa VGA, CGA i MDA/Hercules grefičkim karticama Pojave ovog čipa u sistemima se očekuje već krajem leta, naročito u grafičkim karticama. Cena čipa iznosi $75 za proizvođače, što znači da će se karbce u koje se ugrađuje prodavati za manje od $500

Kontakt adresa S3 Inc, 2933 Bunker Hill Ln Santa Clara, CA 95054,(408) 986-8144, fax (408) 986-1457

NORTONOV NOVI DESKTOP

Novi Norton Desktop for *Windows* poseduje mnoge osobine Macintosh-evog desktopa i uključuje novi fajl-menadžer, dizajniran tako da omogući lakšu upotrebu od sadašnjeg, i set opcija za vraćanje obrisanih fajlova i dijagnostiku hard diska. Novi fajl-menadžer je zamišljen tako da potpuno zameni originalni koji pati od mnogih slabosti tako da su do sada mnogi korisnici radije izlazili u DOS da bi obavili neke rutinske poslove

U novom dekstopu je moguće dizajnirati nove ikone i pndodeliti ih bilo kom fajlu e ikone se mogu postaviti na bilo koje mesto na desktopu Na primer, ikona koja pripada tekstualnom fajlu se jednostavno odvuče na ikonu printera i štampanje započinje Slično Macintoshevom desktopu, novi desktop raspolaže ikonama za sve prisutne diskove i automatski ih pndodeljuje Norton Desktop raspolaže opcijama SmartErase (služi za povraćaj izbrisanih fajlova) i Shredder (bespovratno brisanje celog sadržaja fajla), a u paketu se nalazi i program Norton Disk Doctor Pored toga, tu je i ceo niz konsnih rutina i opcija – rokovnik back-up, i beč-interpreter Po rečima proizvođača, novi paket će biti potpuno kompatibilan sa svim *Windows* 3.x verzijama

Kontakt adresa Symantec, 10201 Torre Ave, Cupertino, CA 95014-2132 408 253-9600, fax (408) 253-4092

TERMINATOR II

U nizu ploča sa procesorom I80486 i radnim taktom od 25 MHz, sigurno da ima razlika. Do sada smo prikazali veći broj takvih mašina (ACS, Miacom/Miakar & Co., Moat, Miadoat), uglavnom sa dobrim rezultatima. U međuvremenu, tehnika je napredovala, cena au padala, a prodavci su marljivo radili na integraciji alatama. Ovaj model predstavlja majstorstvo ovog poslednjeg – integracija.

Dejan V. Veselinović

Isuviše često se misli da svako se šrafciga-rom u ruci može lako i dobro sklopiti računar od nevaznih delova u apsolutnom smislu, to jeste tačno, ali ne kaže mnogo o tome kako će računar raditi kao celina. Sa druga strane integracija sistema nije neka vrsta crne magije, nema bajanja i čarobnih prečica, to je mukotrpan posao koji zahteva poznavanje opreme i puno puno eksperimentisanja. A nagrada može biti velika.

Ovaj model smo dobili na probu od ljubljan-sko-austrijske firme „MiecoM"/"Miakar & Co." Verujemo da ih ne treba posebno predstavljati, jer spadaju u red starih firmi kod nas koje se bave računarima. U telefonskom razgovoru rekli su nam da se sve više okreću procesu integracije kao logičnom narednom koraku razvoja pošto su tržišta obima prodaja zadovoljili, zdrav stav se kojim je teško ne složiti se. Ovaj računar je jedan od prvih koji su prošli ceo proces integracije, pa je za nas bio posebno inte-resantan iz dve razloge. Prvo ovo je prvi put da nam iko eksplicitno kaže da se pozabavio integracijom sistema, na kao nametanjem ne-go kao predlogom kupcima, koji i dalje ostaju slobodni da sami izaberu komponente, pošto kupci to najčešće ne umeju da urade baš najbolje. Imaju savet kako to da urade i izvuku maksimum za svoja para i drugo ceo naš projekat R/Optima se zapravo svodi na integraciju sistema, pa nas je zanimalo da vidimo kako to drugi rade i da uporedimo sopstvene se njihovim rezultatima.

Takođe smo se raspitivali oko izbora diska. Seagate nije baš u milosti. Rekli su nam da znaju za to, ali da su njihova iskustva se ovom firmom u celini dobra (se ogradom za poznate probleme se serijom 137/237) i eve dokle buda tako. Na vide razlog zašto bi na silu boga me-njali (mada po želji mogu da laporuče ili ugrade i druge diskove, ako to tražite). Procenat vra-ćenih nelspravnih diskova se, po njima, kreće u normalnim granicama za većinu firmi, a to što je broj apsolutno veći od drugih zavisi od činjenice da se ovih diskova proda mnogo više od drugih. Opet, teško je naći grešku u ovakvom razmišljanju, posebno kada se zna da je ova firma stari prodavac Seagate diskova i ako ih iko dobro poznaje, to su sigurno oni. Pogle-dajte samo njihove reklame od pre par godina. Najzad, i mi smo u naš projekat R/Optima SX Mk 2 ugradili Saagate disk, uz poznate dobitke na performansama, a do danas nismo imali bi-lo kakvih problema, iako je računar u svakod-nevnoj upotrebi.

TEHNIKA

Računar se osianja na matičnu ploču koja, suprotno današnjim trendovima, ima format prave, velike AT ploče. Za uzvrat ne sebi sad-rži podnožja za svih 16 MB sistemske memo-rije u vidu SIM modula, što smatramo pohvalnim jer eliminiše razne specijalne kartice koji-h često nema, uprkos obećanjima proizvođača. U vezi memorije, još dva stvari malo nas je začudilo da ploča na prima SIM module od 4 MB, odnosno 64 MB u celini, a po svemu su-deći, ni BIOS ne podržava više od 16 MB. Ru-ku na srce, do sada smo videli samo jednu je-dinu mašinu ovog tipe sa svih 16 MB (Miadost, Loznica), all to je bilo na naš izričit zahtev zbog potrebe testiranja AutoCAD a. Za uzvrat ploča trpi mešanje modula od 256 KB i 1 MB a jedni uslov je da na možete mešati module unutar jedne banke (tj. eva 4 SIM-a moraju biti isti). Ovo je opet dobro jer odzvoljava kupovinu se-mlo memorija i naknadna proširivanja bez za-mena prvobitnih modula, mislimo na varijantu kada se zbog štednje prvobitno uzma 1 MB a posla se ustanovi da bi ipak malo više dobro došlo (po našem iskustvu odata tipična situa-cija po sistemu „prako preče nackolo bliže").

Ploču sačinjevaju dve OPTI VLSI kola (82C481 i 82C462), kao i jedno SIEMENS VLSI kolo se oznakom 82C206 koja, koliko znamo, omogućava sinhronizaciju sebrinice i eksterne keš memorije. Kao i svi zreli proizvo-đi, i ova matična ploča ima spoljnu sekundarnu keš memoriju u vidu 16+1 SRAM kola (CY7C185-25PC) 16 komada su seme keš memorija od 128 KB a sedamnaesto kolo je tzv. TAG RAM i služi održavanju koherencije keš memorije. Veoma nam se dopalo to što ploča ima isključivo 16-bitne utičnice za proši-renja, iako se u svakoj mašini nađa po neka 8-bitne kartica, odbro je znati da ne morate de sa sabirete gde ćete šta staviti kako biste izvu-kli maksimum, sve je maksimum.

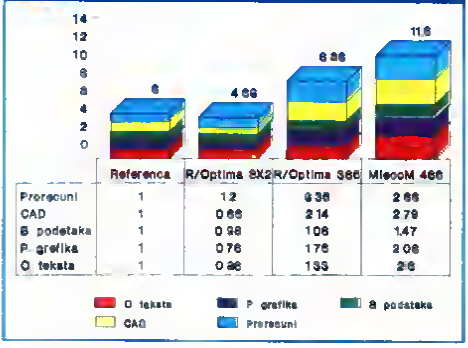
Broj kratkospojnika na ploči je odsta veliki all se korisnik bakće se samo par komada, ostali se podešavaju prilikom inicijalne opreme ploče (recimo dva kratkospojnika određuju ko-ličinu keš memorije itd.) Štampa je uzorno uredna pa čak i lepa ne oko. BIOS potiče od Amerčke firme AWARD i veoma je „taze" od kraja prošle godine, pored toga, izuzetno je do-bro organizovan u smislu podešavanja, daleko ispred ostalih sličnih BIOS-a koje smo videli od firmi kao što su Phoenix i AMI. Pojednostavljen je u smislu da nema suvišnih registara koji ba-caju u očaj sve sem profesionalace (a često i njih), a opet ima eme baš sve što možete po-želeti da izmenite. Sem toga, uz ploču odbijate i dve izuzteno dobro napravljene (napisane, prezentirane i oštampane) knjižice, od kojih jedna služi podešavanju ploče, a druga podeši-šavenjem parametara BIOS-a i CMOS RAM a. U ovoj drugoj smo prvi put u životu videli i slike preporučenog podešavanja, svaka čast! Što je veoma važno, možete podesiti brzinu rada se-birnice, i to kao sistemski takt podeljan sa 4 (6.3 MHz), 3 (8.3 MHz) i 2 (12.5 MHz).

Flopi disk jedinice potiču od manje poznate japanske firme JPN, deluju sasvim solidno, na izgled nismo primetili bitnija razlike između TEAC, Panasonic i drugih čuvenih imena i ovih. Opa, za diskete od 5.25 inča nosi oznaku DS 53A, a ona za diskete od 3.5 inča DS 34A. Lepo rade, standardno izgledaju, a jedino što ih odvaja od drugih je nestandardna brzina reda (mada je u ovom smislu Panasonic i dalje ne-pnkosnoven).

Tvrdi disk je Seagate model ST-1144A u propratnom papiru piše da disk ima 1021 cilin-dar, 7 glava i 30/36/44 sektora. Uprkos tome, i BIOS-u je disk bio definisan sa 1024 cilindra, 14 glava i 17 sektora (tip 48), kako i zašto ne znamo, ali da je radio, vela radio je kao lud.



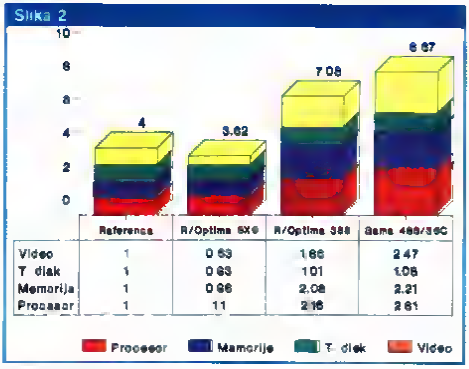
Slika 1



	Reference	R/Optima 8X2	R/Optima 386	MiecoM 486
Prorecuni	1	1.2	8.38	2.66
CAD	1	0.66	2.14	2.79
B. podataka	1	0.98	1.06	1.47
P. grafika	1	0.76	1.76	2.06
O. teksta	1	0.86	1.33	2.6

Miacom 486 – hardverski test

Slika 2



	Reference	R/Optima 8X8	R/Optima 388	Game 486/33C
Video	1	0.63	1.66	2.47
T. disk	1	0.63	1.01	1.06
Memorija	1	0.66	2.08	2.21
Procesor	1	1.1	2.16	2.61

Miacom 486 – softverski test

Sa ovim podacima formatirani kapacitet diska je 124 780 544 bajtova, tj. vem 125 MB. Na izgled sasvim je nalik našem manjem modelu ST-1102A: jedneko je tih i još uvek pomalo neuraden sa onih par islina dobro izolovanih otpornika preko pola ploča sa elektronikom. U celini deluje ubedljivo, a cena mu je dobra.

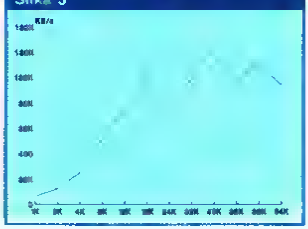
Tastatura je poraklom od tajvanske firme „Chicony". Ne bismo baš smeli da se zakunemo, ali mislimo da je ovo neki novi model jer se razlikuje od prethodnih proizvoda ove firme i to u pozitivnom smislu. Prvo, dosta je solidne, plastika je prilično masivna, tasteri se na klale a nalazi se negde u savršenoj sredini između tvrdih IBM i mekanih KeyTronica i Cherry tastatura. Niti je isuviše mekana, niti isuviša tvrda, oznaka modela je „Chicony KB-5181". Drugo, što je daleko važnije, raspored tastera je prema QWERTY rasporedu, a na cmo označena slova, crvenom bojom su naneseni i naši znaci prame YUSCII standardu, sva to deluje sasvim prihvatljivo, mada baš lepo nije. Ukratko: tastatura je dobar izbor sam ukoliko ne insistirate na potpuno bešumnim.

Video podsistem takode daleko prevazilazi prosek ponude kod nas: bar prema našim saznanjima. Video karte konstr. Tseng ET4000 čip u idealnoj verziji sa šest pojedinačnih oscilatora umesto jednog oscilatora i generatora takta u vidu jednog čipa, tvrdimo da je ovo brža varijanta, a postignuti rezultati na testovima potvrdduju ovaj stav (mada su istini za volju, razlika minimalne). Kao i sveki dobar proizvod u ovoj klasi, kartaje imala ceo megabajt video memorija.

Monitor nosi oznaku „Mitac AL-4020". Po mnogo čemu je nalik većini drugih sličnih monitora, ali ima i niz razlika. Na prednjoj strani nosi natpis „Low Radiation", naš metar za merenje statičkog elektriciteta je ostao potpuno mrtav (ček i na 1 cm od ekrana, što nikada do sada nismo doživeli (ček smo proveravali bateriju, jer smo mislili da je ispustila dušu). Ček ni se odličnim Philips Brilliance monitorom. Jednostavno, ovaj monitor toliko malo zrači da bi bilo potrebno imati daleko skuplju i precizniju mernu opremu. Alal vera!

Premaz ekrana konsti p22 fosfor sa veličinom tačkice od 0.28 mm. Tasten za uključivanje, osvetljenje i kontrast se nalaze na prednjoj strani, a sa leve strane postoji jedna klapnica ispod koje se nalaze dopunske komande za horizontalnu fazu, veličinu po vertikali i vertikalno centriranja. Sasvim dosta komandi a u praksi nam ioneko nisu zatrebale.

Slika 3



Seagate ST 1144A – brzina prenosa

MERENJA

Ceo računar smo proteral kroz naše uobičajene testove i bili prilično iznenađeni dobijanim rezultatima, koji su podaleko iznad proseka ova klase. Tačnije rečeno, ovo su do sada najbolji rezultati sa izuzalkom Game 486 sa EISA sabirnicom i na rednom taktu od 33 MHz. Komentar bismo podelili u dva grupe: opštu i posebnu.

U okviru opštih komenatara, prilično je jasno da se radi o sada već zrelo] tehnologiji koja sa nelazu u svom trećem izdanju. Sav taj razvoj je dao svoje razultate kroz veoma integrisanu lahniku koje je same po sebi usled integracije dosta brza, pe u saradnji sa prirodno brzim procesorom daje dobro integrisanu celinu.

Što se posebnih komenatara tiče, njih bismo opet razveli u dve grupe. Prvo što nas je veoma iznenadilo su rezultati testove sa tvrdim diskom koji su blago rečeno izvanredni. Na testovima u radi sa bazama podataka dobili smo indeks od ček 1.1.47, vaoma neubočajeno visok razultat. Zamenom Seagate dis'e za Quantum disk razlike sa pokazala u nižem nivou performansi. Istina za samo malo, ali kada se uzma u obzir cena, prednost Seagate-a je jasne. Slično važi i za razultata na video testovima: ova kombinacija je u tekatualnom režimu dobro prebacila ček i Compaq-ova razultate, posiovično najbolja na tržištu. Zapravo da i nema skrolovanja, odjednom je sve tu i jedva sa nazira kretanja teksta.

Druga stvar je integracija celine. Očigledno je da je cell sistem odlično usaglašen, delovi su dobro odabrani a sve je podešano kako valja, da se dobiju odlični razultati. Mogle bi se reći jedino to de ovakve mašina ipak zaslužuje veći disk, recimo od nakih 170-210 MB (a kao server još mnogo veći disk, od 330 nagora), mada prema performansama, ovome sa teško našto može zameniti. Ideje da se sve ovako ukomponuje se očigledno pokazala dobrom, kao i sama izvedba: de je brzo, brzo je, de ja najbrža u klesi 486 na 25 MHz, najbrže ja.

MERENJA

Ceo računar smo proteral kroz naše uobičajene testove i bili prilično iznenađeni dobijanim razultatima, koji su podaleko iznad proseka ova klase. Tačnije rečeno, ovo su do sada najbolji rezultati sa izuzalkom Game 486 sa EISA sabirnicom i na rednom taktu od 33 MHz. Komentar bismo podelili u dva grupe: opštu i posebnu.

U okviru opštih komenatara, prilično je jasno da se radi o sada već zrelo] tehnologiji koja sa nelazu u svom trećem izdanju. Sav taj razvoj je dao svoje razultate kroz veoma integrisanu lahniku koje je same po sebi usled integracije dosta brza, pe u saradnji sa prirodno brzim procesorom daje dobro integrisanu celinu.

Što se posebnih komenatara tiče, njih bismo opet razveli u dve grupe. Prvo što nas je veoma iznenadilo su rezultati testove sa tvrdim diskom koji su blago rečeno izvanredni. Na testovima u radi sa bazama podataka dobili smo indeks od ček 1.1.47, vaoma neubočajeno visok razultat. Zamenom Seagate dis'e za Quantum disk razlike sa pokazala u nižem nivou performansi. Istina za samo malo, ali kada se uzma u obzir cena, prednost Seagate-a je jasne. Slično važi i za razultata na video testovima: ova kombinacija je u tekatualnom režimu dobro prebacila ček i Compaq-ova razultate, posiovično najbolja na tržištu. Zapravo da i nema skrolovanja, odjednom je sve tu i jedva sa nazira kretanja teksta.

Druga stvar je integracija celine. Očigledno je da je cell sistem odlično usaglašen, delovi su dobro odabrani a sve je podešano kako valja, da se dobiju odlični razultati. Mogle bi se reći jedino to de ovakve mašina ipak zaslužuje veći disk, recimo od nakih 170-210 MB (a kao server još mnogo veći disk, od 330 nagora), mada prema performansama, ovome sa teško našto može zameniti. Ideje da se sve ovako ukomponuje se očigledno pokazala dobrom, kao i sama izvedba: de je brzo, brzo je, de ja najbrža u klesi 486 na 25 MHz, najbrže ja.

ZAKLJUČAK

Cena ovakvih matičnih ploča se već dovoljno spustila da im omogući da postanu veoma dobar izbor za poslovna mašine jakog kalibra, dakle za jači rad sa CAD-om, kao server i tome slično. U svim tim podvarijantama trebaće još memorije, jer se osnovni modei isporučuja sa 4 MB, a mi smo dodali još toliko. Kao server, ovaj računar bi morao da ima veći disk, ali to po svemu sudeći nije neki problem, prostom zamenom sa lako može postići sve što treba.

Kao server, ovaj računar bi mogao činiti čuda sa nekim od savremenih inteligentnih kontrolera, jako je ograničen na ISA sabirnicu, po našem iskustvu. EISA je ipek više obećanje svetla budućnosti nego realnost sadašnjice. Provera radi, odvojili smo prvo 1 pa 2 i najzad 4 MB kao keš memoriju diska i sasvim očekivano dobili razultate od kojih se ostaje bez daha (nismo ih prikazivali, all recimo indeks testa sa bazom podataka sa 1.47 raste na 6.6, odnosno za 4.62 puta). Da smo u testu sa CAD-om koristili verziju programa koja podržava koprocesor, indeks od 1.96 bi porastao na 3,33, i tako dalje.

Pošto je ovaj računar pokazao osetno bolje performanse nad našim modelom R/Optima 486 i pošto smo već izmanili našu referentnu mašinu, prihvatili smo ovaj računar kao novu referencu ze visoku klasu, ovo je dakle R/Optima 486 Mk.2. Čudo jedno šta malo rada na integraciji može da napravi.

Sve u svemu, impresivna mašina što zbog same lehnike, što zbog odlične integracija delova u celinu. Toplo preporučujemo.

Kontakt adresa:

MiecoM d o o
Celovška 185, 61000 Ljubljana
Tel 061-556-484, Fax 061-556-485
Miakar & Co., Export-Import
Unterbergen 62, Austrija
Tel 9943-4227-2333, tax 9943-4227-2091

SX: NOVA GENERACIJA

O pločama sa procesorom SX smo do sada dosta puta pisali – čak smo i napravili projekat R/Optime SX, sada vec u Mk.2 fazi. U meduvremenu, dosta toga se dogodilo – pojavila se i sasvim nova generacija matičnih ploča sa ovim procesorom, stigle su nove ploče sa procesorom 80286, a došlo je vreme i da se Unitest pomalo osveži – eto razloga za ovaj članak.

Dejan V. Veselinović

Razvoj tehnologije integracija je doveo do pojave najnovije generacije AT matičnih ploča koja se svode na faktički jedan jedini čip, poreklom od firme Headland sa oznakom H12. O njemu smo već govorili prilikom testa računara preduzeca „Šuli-ko" u onoj verziji ploča je radila na 25 MHz. Medutim, ništa manje je značaj na pojava isle takve ploče na 16 MHz dakle u narodnoj klasi koja nudi neke interesantne vili ne. Pored očiglednih kao što su visoki stepen integracije velika pouzdanost CMOS tehnologija i slično najvažnija prednost je rad sa memorijom

NOVA TEHNOLOGIJA...

Usled visokog stepena integracija (sve u jednom čipu) postalo je lako izvodljivo da se sve prednosti precizne sinhronizacije celokupne logike izvedu veoma lačno i što je još važnija da se tajno održe u sinhronizovanom ta žimu jer se cela logika nalazi na istom substratu istog čipa. Ovim se postiže da sa svi vremenski gubici unutar samog procesora mogu lepo iskonstili u spregi sa memorijom kako bi se dobio rad bez ikakvih slanja čekanja (brzina procesora je ograničena jedino šinnom magistrale i radnim taktom)

Tako postaja moguće uprkos matematici koja za takt od 16 MHz bez slanja čekanja na vodi brzinu memorije od 62.5 nS radili bez ikakvih slanja čekanja i se memorijom od 70 nS koje ima kolonia hocela i gde hočete jer se na početnu ciru dodaje vreme koje procesori potroši unutar samog sebe. To je nakih 8.5 nS kada to dodamo na 62.5 dobijamo potrebnu brzinu od 71 nS

Ovo je naravno bilo poznato odavno. Razlika izmedu onda i sada je u tome da je integracija u celosti uklonila sve probleme oko sinhronizacije i sto je još važnije oko odižavanja postignutog nivoa sinhronizacije na dugi rok. Za nas pojava koju je trebalo provaliti u praksi, što smo i uchili

Na frontu narodne 32-bitne klase odigrale su se dva značajna događaja. Prvo AMD je preločio svoj klon procesori 386 u SX varijantu, a za razliku od Intela odmah ih je ponudio na radnim taktovima od 20 i 25 MHz (Intel nudi samo do 20 MHz). Drugo ogromno širenje fronta prodaje 32 bitnih mašina je dovelo do masovne upotrebe statičke memorije i velike razlade pomoćnih kola pa proizvodačima nije bilo teško da kola namenjena procesoru 80386 prilagoda manjem bralu zajedno sa keš kontrolerima

Jako tehnički nema nikakvih problema raditi na 20 pa i na 25 MHz koristeći preplitanje i lako dostupnu memoriju isto lako nema problema ni oko ugradnje keš memorije koja isti na dodaje nešto na cenu ali sigurno i na performanse. Ekonomska klasa od 32 bita dakle dobija ubrzanje u vidu nova generacije matičnih ploča sa procesorom SX na višem radnom taktu i sa keš memorijom E to već nije moglo da proda pored nas bez probe

...STARI CILJEVI

Prvi i osnovni cilj nam je bio da isprobamo nove proizvode. To je bar jasno jer se ipak radi o novim „standardnim" proizvodima koji se nude na našem tržištu

Drugi cilj nam je bio da uporedimo te nove proizvode sa dosadašnjim standardima kako bismo videli i utvrdili o kolikim razlikama u performansama i cenama se radi. Ako performanse porastu recimo 10% a cene recimo 20% onda baš i nismo nešto dobili, no ako cene ostanu isle a performanse porastu onda jasno dobili ma kolika razlika u performansama bila

I treće mada odmah priznajemo da je to bila slučajna sinhronizacija došli smo u fazu kada je trebalo ponovo malo preliesti naš sistem merenja „Unitest". Tu su i novi mašinski testovi neke izmene u sistemu merenja i neki malo doteranji testovi. Kada je već sve to nekako tu, pomislili smo da bi bilo veome zgodno da varzija 3.1 „Unitesta" eventualno dobija i novu referencu

DVE KONFIGURACIJE

Prvi računar se sastojao od jednog Baby AT kucišta standardne flop disk jedinice od 1.2 MB standardne Hercules klon karte sa Philips žutim monitorom i standardne AT tastature sa 101 „klik" tasterom. Tvrdi disk je Quantum 52A formatiranog kapaciteta od 52 MB a matična ploča vec pomenuta Headland ploča na 16 MHz sa 1 MB RAM-a (Siemens 70 nS)

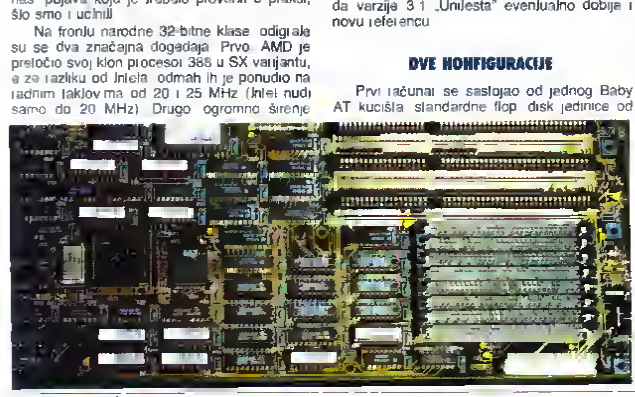
Drugi računar je u celini potpuno isti sem naravno u slučaju matične ploče koja je u ovoj verziji jedna SX ploča na 20 MHz sa 4 MB RAM-a i sa 16 KB SRAM keš memorije. Ploče koriste OPTI čipove (82C281 i 82C206) a SRAM čipovi nose oznaku „Intel" sa brzinom pristupa od 35 nS. Pored ovoga, ploča je interesantna i zbog toga što se prva 4 MB memorije slavljaju u vidu SIM modula a ostalak (dalji 4 MB) u posebnim postoljima koje od sada uživo nikada nismo videli mada znamo sa slika da AST koristi ta ista šina SIM postolja već poodavno. Pomalo se pitamo gde ti ce sa naci memorija za ta postolja. mi je nikada nismo videli ali ako sa masovnije klena sa njima valja da ce se pojaviti i čipovi

Obe ploče koriste AMI BIOS. G2 ploča ima 4 16 bitne i 3 8-bitne utičnice a SX ploča 7 16 bitnih i 1 8 bitnu. Obe su veoma uredno napravljene bez ikakvih žica i sa čak veoma lepom štampom. ovo se posebno odnosi na SX ploču koja je izuzalno uredno napravljena (tako to nigde ne piše čini nam se da se radi o proizvodu Jajvanske firme BSI)

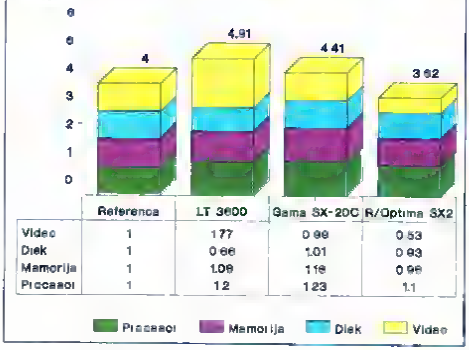
NOVO NA HEADLAND PLOCI

Svi rezultati testova a naravno posebno merenja memorijskih testova jasno pokazuju da ove ploca zaista radi bez ikakvih stanja čekanja (nema preplitanja tzv „true zero wait state"). Rezultati koje smo dobili na procesorskim operacijama su stvarno bolji od dosadašnjeg standarda. NEAT ploče na 16 MHz sa dvostrukim preplitanjem. Ili lačnije 10-15% bolji. Što je najlepše. Ta razlika u brzini je postignuta uz smanjenu cenu naime razlika u ceni ove ploča u odnosu na NEAT ploču je 35%. Tako se došlo do one najpožaljnije moguće situacije plaćate apsolutno manje za apsolutno bolje performanse! Što je najlepše ne žrtvujete ama bas ništa i ovde postoje sve one lepe stvari za podešavanje (duplikalna memorija za BIOS i/ili za video prebacivanje 384 KB iznad 1 MB) all uz dodatu prednost lakoća podešavanja

Ovaj računar kao celine odnosno kao sistem je podigao referentni nivo za poprilican korak iz dva razloga. Prvi je efikasnost procesora i memorija mada moramo reći i veoma dobrog AMI BIOS-a koje je sada kao nikada ranije integrisan u jednu veoma kompaktnu celinu. To su zapravo odlike veoma zrele tehnologije. Drugi razlog opšlem porastu performansi je naravno Quantum disk koji je blago rečeno odličan ko ne veruje neka proveri. U saradnji sa pločom opšti nivo performansi je porastao za procenih 14% što uopste nije malo za već poznatu tehnologiju



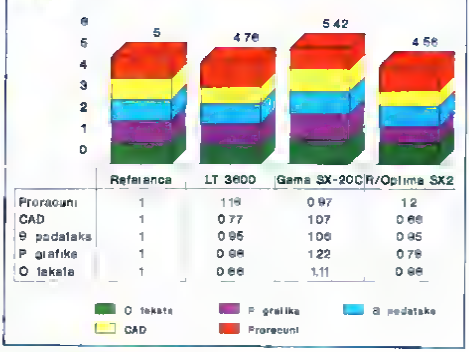
Slika 1



	Reference	LT 3600	Gama SX-20C	R/Optima SX2
Video	1	1 77	0 99	0 53
Disk	1	0 66	1.01	0 93
Memorija	1	1.09	1 19	0 99
Procesori	1	1 2	1 23	1.1

Gama SX 20C – hardverski test

Slika 2



	Reference	LT 3600	Gama SX-20C	R/Optima SX2
Proracuni	1	1 19	0 97	1 2
CAD	1	0 77	1 07	0 66
B podataka	1	0 95	1 06	0 95
P grafike	1	0 66	1 22	0 79
O teksta	1	0 66	1.11	0 66

Gama SX 20C – softverski test

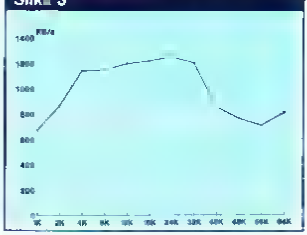
NOVO NA SX PLOČI

Kao što rekosmo ova konfiguracija se razlikuje od prethodne samo po matičnoj ploči u ovom slučaju procesor radi na zdravih 20 MHz sa 16 KB keš memorije i 4 MB sistemske memorije. Uzgred, keš memorija je proširiva (zamenom čipova na podnožjima) na 64 KB a već se nejavljuju slične ploče na 25 MHz i keš memorijom od 32 do 256 KB (!). Medutim 8 KB koliko je koristila ploča firme Compaq je dovala do osetnih razlika u performansama, a podsećamo na staro pravilo poznato od ranije da je algoritam keš memorije daleko važniji od proste količine.

Naravno, ova ploča je zahtevala da se uporedi se R/Optimom SX Mk 2 koja takođe radi na 20 MHz ali nema keš memoriju sa R/Optimom 386 Mk 3 koja je najnovija inkarnacije ovog projekta (a razlikuje se od Mk 2 samo po disku. Fujitsu je ustupio mesto boljem i bržem Quantum disku 170A) i naravno našoj referenci u portabilima Chicony LT 3600 (NEAT ploča na 20 MHz sa čelvorostraničnim preplitanjem). Ovaj poslednji model je već pomalo ograničen svojim Conner diskom koji pokazuje znake vremešnosti u brzini pristupa (24 mS Quantum 52A 16 5 mS ili 45% brže) pa čak i u domenu svoje najveća prednosti organizacij i iz vedbi madumemonje na semom disku. Conner je tu uvek bio car ali vremenom ga je Quantum sustigao a bogami i prestigao.

Sve u svemu ova ploča je dosla rafiniran proizvod koji jasno pokazuje znake svoje zrelosti od izgleda do performansi jasno se vidi da proizvođač ima poduže iskustvo sa ovakvim proizvodima. U odnosu na R/Optimu SX Mk 2 keš memorija je popravila performanse za 11% ne čisto procesorskim testovima odk je ukupna razlika u performansama nešto veća (21 1%) ovaj porast potiče od objektivno bo-

Slika 3



Gama SX 20C – brzina prenosa za tvrdi disk Quantum 52A

TESTOVI

NOVI UNITEST

Dalji razvoj našeg „Unitesta" je doveo do još jedne verzije, koja jeste jasan korak napred, ali ne toliki da zavredi oznaku potpuno nove verzije. Izmena i novosti ima nekoliko.

Prvo, neki mašinski testovi su znatno osavremenjeni, a uveden je potpuno novi test za procesore iz porodice 486, i SX i DX. Drugo, neki testovi su značajno prerađeni u smislu radnog modela koji koriste, i to uglavnom ka većem stepenu složenosti, pošto mašine postaju sve brže i brže, neki testovi su postali nepouzdano kratki, pa je trebalo produžiti vreme izvršavanja kako bi i rezultati postali pouzdaniji.

Treće, promenili smo referentni model, koji je opisan u priloženom članku. Na ovaj korak smo se teško odlučili, jer je veoma nezgodno menjati baš referencu, pošto se time ceo sistem vrednovanje baca u haos. Ipak uradili smo domaći zadatak pa ako poželite da stare rezultate dovedete na ovaj novi od danas važeći nivo vrednost reference pomnožite 1 11 odnosno rezultate drugih mašina podelite sa 1 11. Nije savršeno ali je dosta blizu.

Najzad iako smo preši ne sistem prikazivanja rezultata kao u ovom članku detaljne listinge sa rezultatima svih pojedinačnih merenja od sada pa na dalje ćemo ostavljati na „Sezamu" u konferenciji PC HARD. Neki su se bunili protiv njih neki su ih zahtevali a svi su nas aludeli zalo ne papiru malo uopštenije a ko hoće sve pikantne detalje neka ih preuzme sa „Sezama" A ako nema modem zašto ga nema?

Korisna adrese

GAMA Electronica Beograd
11000 Beograd, Milšarska 11
Tel 011/332 275 Fax 011/335 902

Ijeg (bržeg) diska i činjenice da Hercules karte u nekim testovima rade brže od VGA karti. Naravno rezultati testa u CAD u sa Hercules kartom su apsurdni jer niko pametan ne bi seo da radi CAD sa Herculesom ali princip je princip mora i taj test. Kada se vrednosti konguju za razliku u video stvama razlika u brzini između R/Optime i Game SX je oko 14% što zbog keša što zbog diska.

Interesantno je primetiti da je razlike u brzini između Chicony LT 3600 i Game SX veoma mala jedva 3% u procesorskim testovima naravno SX po delinciji ne može što i ta kako treba uzeti u obzir prilikom razmatranje. Gama SX može da koristi sve prednosti 32 bitne tehnologije a mi prvi na bismo znali kako bismo danas bez svih tih lepih stvari na koje smo se lako lako navikli.

DO SLEDEĆE ITERACIJE

Ulazni nivo današnjice je zaista prijatno iznenadio sve nas koji smo se ovom prilikom sa njim susreli. Teško je da nečemu cena padne uz istovremeni porast performansi i pouzdanosti (bar teoriski). Headland matične ploče je zaista dobar početak na oko 6 5-7 puta višem nivou nego prapočec u vidu IBM PC a za sva ona koji ne mogu bolje (!) na mogu odmah na 32 bitnu tehnologiju ili kojima treba obrada teksta bez ludovanja. Mirne duše preporučujemo ovaj proizvod a nude ga mallene svi prodavci danas. Samo pazite na disk koji uzimate uz mašinu. Quantum jeste dobar verovatno najbolji u ovoj klasi ali je skupiji od proseka. Mi bismo se ipak za njega opredelili.

Gama SX je zreo proizvod koji je podigao performanse ulaznog modela (mada i Gama i drugi prodaju i staru generaciju SX ploča ne 16 MHz bez keša) na jedan zaista razuman nivo performansi u kontekstu savremenih programa a posebno programa koji koriste *Windows 3.0*. Ovih skromnih 16 KB keša je sasvim dovoljno za rad u pristojnom okruženju odnosno sa razumnom brzinom koja je poređenja radi svojih 40-50% veće od starijih modela ploča na 16 MHz koje su po nama prespora. Čak i pod *Windows*om brzina rada sa ovom pločom a posebno u sprezi sa brzim Quantum diskom je već sasvim dobra. Mirne duše dobija preporuku.

Bar do sledeće iteracije SX na 25 MHz sa 256 KB keša

PROCESOR KOGA JE TEŠKO UBITI

Priča o procesoru intel IAPX 80286 počinja vaoma davno – u praistoriji ličnog računarstva. Čima je prikazala svoja nova, i za ono vrama izuzatno napradna procesore iz aarija 8086/8088, amarička firma Intel je gotovo odmah otpočala rad na njihovoj zamani. Osnovni cilj prva faza razvoja je bio vači atapan integracija parifarnih kola u okvirima samog procesora. Tako su rođani procesori IAPX 80186/80188, saavim nalik svojim prathodnicima, ali brži uslad veča integracije; ova priča sa ponovila sa procesorima 80386 i 80486.

Dejan V. Veselinović

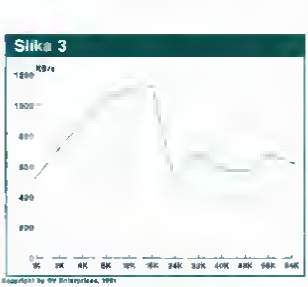
No ovo je bila samo odskočna daska za stvarni korak napred, nazvan tada iAPX 80286. Za razliku od svojih prethodnika, ovaj čip može da adresira na 1 nago 16 MB memorije, porad standardnog režima rada, nazvanog „realni režim" (real mode) u kome on glumi (emulira) brzi procesor 8086 ili 80186, ubečan je i posaban režim, nazvan „zaštićeni režim" (protected mode). U ovom drugom režimu, procesor je sposoban da pristupi svim 16 MB memorije, mada na žalost i dalje u segmentima od po 64 KB, mani koju je nasledio od prethodnika.

Isto tako na žalost, ovaj procesor je imao i jednu ugrađanu manu, kada ga jednom prabacila u zaštićeni režim, nema povratka nazad bez potpunog rasetovanja računara. Sve ostalo na stranu, a bilo je dosta priča koja su potekle i iz konkurantskih labora, ali ovo je vaoma velika mena koja je neosporne.

No, uprkos svemu, a posebno pojavi naprednijih 32-bitnih procesora, ovaj čip je opstao do dan-danas. Glavni razlog za taj uspah je činjanica da se on ipak koristi kao brzi 8088, jer i dalja služi kao motor za stan aulo pod imenom DOS. Dalje, ovaj procesor je doživeo do sada nevidanu avoluciju, bez presedena u istoriji mikroprocesora, sa skromnog radnog takta od 6 MHz, dogurao je do 4 puta vačeg takta: 25 MHz, a ida se i dalje od NMOS tehnologija, prabačen je na submikronsku CMOS tehnologiju, opet veoma radak potez, i što je najvažnije, imao je više alternativnih izvora pored samog proizvođača, firma Intel. I dalja ga prave i razvijaju firme Harris, AMD, Siemens i neke druge, a sva ta konkurancija je dovala do pada cene samog procesora, ali i razvoja pomoćnih kola za njega.

Ukratko, ovo je procesor koji jednostavno na dozvoljeva da bude ubijan čak ni od strane njegovih tvoraca. Intel-a, koji ga više ne proizvodi, sem za posabne namana.

Njegova cena i rasprostranjenost su dovali do toga da se i danas razvijaju posabna vezna kola njega, poslednja u tom nizu je kolo od firma Headland Technologies, koja ispunjava stari san o celom AT računaru u jednom čipu (na baš sasvim, ali skoro). Naravno, ni drugi se na daju. Zato smo ovaj put proball dva računara koji su različiti a opet slični, jer su obe namenjena onima kojima treba natprosečna brzina rada uz skromniju canu.



Brzina tvrdog diska (Šutlić)

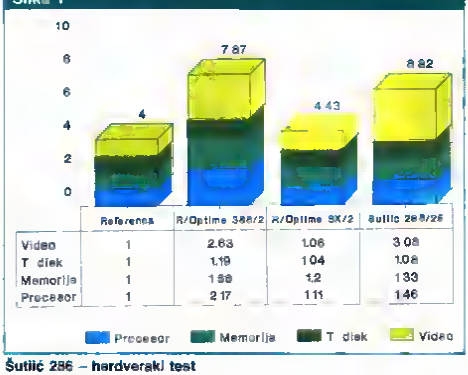
GENELEC GA286

Beogradsko-tuzlanska firma Ganalec vam je možda i poznate iz reklama, ali mi do sada nismo sem lih reklama imali bilo kakvog dodira sa ovom firmom i u tom smislu, GA286 nam je prvi konkretan kontakt se ovom firmom.

Samo kućišta računara je veoma interesantno. Radi se o jednoj inkarnaciji tzv. „slim line" kućišta, odnosno kulija koja su osalno smanjanih dimenzija u poređenju sa uobičajanim Baby AT kulijama. Međutim, ova kutija nija ništa bitnija uže niti plića od standardnih, ali je zalo osalno niža, otprilike polovina visina nekog uslovnog proseka. Ideja projektanata je bila, po svemu sudeći, da kada stavite monitor na kutiju i ako sve zajedno stavita na prosečno visok sto (70-80 cm), monitor bude otprilika u ravni sa vašim očima, sem toga, cala kutija izgleda kao nekakva „radna stanice" (za ljubitelja HiFi ja „profi look").

Da bi ava to bilo moguće, moralo se žrtvovati mnogo toga. Prvo, proširenja u obliku kartlica moraju biti postavljena horizontalno a ne vertikalno, što se postiže ubacivanjem jedne dopunske kartice sa utičnicima u jednu standardnu utičnicu, kako bi se promanio ugao. Ovim se ujedno broj utičnica svodi na četiri: 3 16-bitna i jednu 8-bitnu, u našem primerku ma šina, sve četiri utičnice su efektivno bile popunjana. U jednoj se nalazila VGA karta, u drugoj AT Bus adapter, u trećoj I/O karta, a procep na zadnoj strani kulija za četvrtu kartu je bio zauzet sa dva RS 232C veznika (mada je sa ma utičnica prazna).

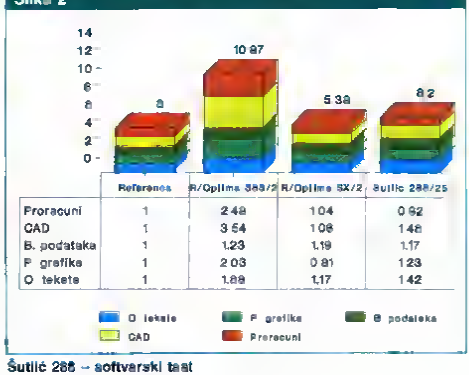
Dalje, pošto su i fiopi disk jedinica i tvrdi disk poslavljani jedno kraj drugoga, najveća mogu ća dužina kartica je dvotrećinska, o punoj duži



	Reference	R/Optime 386/2	R/Optime SX/2	Šutlić 286/25
Video	1	2.63	1.06	3.08
T. disk	1	1.19	1.04	1.08
Memorija	1	1.88	1.2	1.33
Procesor	1	2.17	1.11	1.46

Procesor, Memorija, T. disk, Video

Šutlić 286 – hardverski test



	Reference	R/Optime 386/2	R/Optime SX/2	Šutlić 286/25
Proračuni	1	2.48	1.04	0.92
CAD	1	3.54	1.08	1.48
B. podataka	1	1.23	1.19	1.17
P. grafika	1	2.03	0.81	1.23
O. tekste	1	1.88	1.17	1.42

O. tekste, P. grafika, B. podataka, CAD, Proračuni

Šutlić 286 – softverski test



Snimio: Nenad Marijanović

Šutlić 286/25

ni ni na razmišljajte. Najveća dužina kartica bi bila i manja (poludužinska) da tvrdi disk nije Maxtor sa prečnikom od 3,5 inča, se odgovarajuće manjom dubinom. Sama matična ploča je opet sasvim standardnog oblika i veličine. Mala visina je zahtevala da ventilator za hlađenje bude na dnu kutije i da vazduh gure nadole, u prostor od oko 4-5 mm od površina stola. Loše rešenje.

Pošto se zadnje strane kutije postoji jedan posaban, izuzetno širok prosek jasno je da je kutija namenjena posebnim matičnim pločama koja na sebi sadrže sve potrebne veznike i AT Bus adapter (a možde čak i video podsistem). Da je to tako, tri od četiri utičnice bi bile potpuno prazne, čima bi bilo omogućeno dalje proširanje računara. Oveko nemate više kude. Dalje, da su tvrdi disk i flopi disk jedinica pomereni udesno za onih 5-6 cm koliko sada ima prazog prostora (ili da je tvrdi disk postavljan ispod flopi disk jedinice), dve 16-bitna utičnice bi mogla da prima kartice pune dužina. Čima bi se otvorio put ka proširanjima. Ovako, na žalost, kutije je potpuno van sinhronizacija sa elektronikom, čak toliko da vam toplo savetujemo da od Genelec-a zahtevate standardnu kutiju. Sa ovom realno na dobijate ništa, a gubite maltene sve (u smislu proširanja).

Što je najgore, kvalitet izrade same kutije je odličan: poprečno je podbočena, ni melo se na savija pri prenosu, lim je dosta debeo i potpuno pocinkovan, a sve to znači i da je nužno dosta skupa. Žali bože truda i pare.

Matična ploča je sasvim standardna NEAT ploča sa čipovima deklarisanim na 16 MHz, iako ovo nije prvi put da vidimo ovaj oblik štednje (čipovi na 16 MHz „navijeni" na 20 MHz) ni ranije se nismo slagali sa ovom praksom, pa ni sada. Opasno je čipove terati iznad deklaracije. Može da primi 1, 2 ili 4 MB u obliku DIP čipova, i još jednom toliko u obliku SIP modula. AT bus adapter koristi Western Digital VLSI i nekoliko TTL kola, a VGA karta ima na sebi 512 KB video memorije i Oak Technologies video čip i BIOS. I/O karta nudi gama Centronics i dve RS 232C linije. Naš primerak je bio opremljen sa 1 MB RAM-a.

Tvrdi disk je od firme Maxtor, koje se sva više javlja kao nekakav nezvanični standard. Radi se o modelu 7040AT koji ima kapacitet od 42 MB nakon formatiranja. Flopi disk jedinica je poreklom od firme YE data, modem YB-360B.

Jedini problem koji smo imali sa ovim računarom se odnosio na VGA pnkaz u programu Hervard Graphics: slika je bila potpuno izobličana („razvučena" po vertikali i po horizontali) tako da se meniji nisu ni videli. Radi se o problemu sa Oak Technologies kartama koji nam je i od ranija poznat, međutim, ovaj problem se javlja samo na ovim i ni na kojim drugim kartama, pa ga zato pripisujamo video čipu. Istini za volju, ni u jednom drugom programu nismo imali problema.

Performanse ovog računara su nas prilično naprijatno iznenadile, iako radi na 25% većem taktu od reference, ukupna posečna razlika u brzini je svega 15%, a na softvarskim testovima je uopšte ni nema (ova testove smatramo pouzdanijom indikacijom brzina). Malo smo se zainteresovali i došli do zaključka da je video karta glavni krivac ovom posustajanju. Kónsti elektroniku za sada najjeftinijeg proizvođača na Tajvanu, a evo i objašnjenje zašto je najjeftiniji, koliko para toliko muzika.

U normalnom radu nismo imali bilo kakvih problema sa ovim računarom. Uprkos našem skepticizmu, varijanta sa ventilatorom okrenutim nadole sasvim lepo održava normalnu temperaturu u računaru, mada se neka kola griju malo više no što bismo voleli da vidimo (25% veći takt). „Tipro" tastatura nas nikada nije oduševljavala, ali je sasvim lepo radila. U radnom smislu, nemamo nikakve zamerke.

Sva u svemu, ovakav kakav je, ovaj računar vam na bismo preporučili bez ikakvih obzira na cenu mogućnosti širenje su gotovo nikakva, kutija u celosti na zadovoljava (sem po izgledu), funkcionalne kritanjume, a performanse su ispod realnih očekivanja i mogućnosti. Ovo nije loš računar, ali ja prodavcu ostalo još mnogo rada na razvoju, da bi ga tak nakneđno doveo na ono što nija a trebao bi da buda.

ŠUTLIĆ 286/25

Već po imenu vidite ono najvažnije. Računar potiče od beogradskog preduzeća Šutlić, koristi procesor 80286 na radnom taktu od 25 MHz. Nema sumnje da je ovo do sada najbrža verzija ovog procesora koja je ikada napravljena, a prema izjavama firmi Harris i AMD (ovaj je Harris ov), granice razvoja je 33, pa čak i 40 MHz (ovo drugo uz promenu arhitekture, odnosno samo uz pomoć submikronske tehnologije).

Elem, sama matična ploča je poslednja generacija takvih ploča kako po dimenzijama tako i po upotrebljenoj tehnologiji. Dimenzije su veoma nalik na danas već standardne G2 ploče, ali umesto dva Hedeland Technology VLSI (G2) čipa, ovde se nalazi samo jeden, to je poslednji korak u daljem procesu integracija ovakvih ploča. Pored tog jednog VLSI čipa, na ploči se nalazi još svega devet TTL čipova, procesor (Harris 80C286-25), ROM BIOS i prostor za 8 MB memorije u vidu SIM postolja. Dakle, osnovnu veličinu ploče zapravo određuju same utičnice za proširenja, kojih ima sedam (jedna 8 bitna, šest 16-bitnih).

U našoj konfiguraciji dve 16-bitne utičnice su bile popunjene, a sve ostala prazna, u jednoj se nalazila jedna VGA video karta, a u drugoj kombinacija I/O karte kontrolara tvrdih diskova i flopi disk jedinice. Nismo ljubitelji ovakvih kombinacija, ali moramo priznati da je sve



Snimio: Nenad Marijanović

Genelec 286

ispo radilo. Video karta koristi Trident TVGA 8900B video čip, glavnog konkurenta poznatom Tseng ET4000 čipu, kao i većina dobrih proizvode današnjice: video karta ima 1 MB video memorija, a sposobna je da radi do razolučije od 1024×768 sa 256 boja. Na matičnoj ploči se nalazilo 4 MB RAM-a u vidu četiri SIM modula (TI, 80 nS).

Brzina osvežavanja ekrana u standardnom VGA režimu je 60 Hz, što nije baš izuzetan rezultat u režimu 800×600 sa 256 boja: osvežavanja ekrana se obavlja sa 56,3 Hz, a u režimu 1024×768 sa 86,7 Hz. Ova poslednje brojka je odličan rezultat, ali sve ispod 60 Hz se more amatreti neodgovarajućim, utoliko pre što je ovo funkcija video karte.

Merenja performansi su dala uglavnom dosta očekivana rezultate. Posebno jaka strana ove mašine su video performanse koje rezultiraju od upotrebe duplikatne memorija (shadowing) i činjenice da video čip Trident 8900 na ilk čipu Tseng ET 4000 sadrži internu keš memoriju. Razultat je veoma brz video podsistem, mada se ta brzina ogleda u tekstualnom režimu, dok u punom grafičkom režimu ova karta i dalja zaostaje za konkurencijom se Tseng čipom.

Iako jaste brži, ovaj računar pokazuje razlike u brzini u poređanju sa našom referentnom NEAT mašinom (na 20 MHz) koja su manje od razlike u tektu. Stvarna razlika u brzini (kada se uzme u obzir da reference ima koprocesor a drugi računar ne) je 15,8%, a čak i ta razlika je pod ubcajem brzeg tvrdog diska. Ipak, da je brz, brz je, nema diskusije.

U odnosu na R/Optimu Mk III (najnovija izdanja, zamenjen samo disk, sada umesto diska Fujitsu od 180 MB ima Quantum od 170 MB), naravno da je R/Optima „povukle", a glavna razlika se ogleda u činjenici da brza 32-bitna mašina ima i koprocesor IIT, što joj omogućava nesrazmerno bolje rezultate u tabelarnim proračunima. Uprkos tome, slično i u proporciji sa cenom, i performanse su bolje.

Dakle, imamo ovakvu situaciju: odličan tvrdi disk, dobra video karta, brza matična ploča, pristojne tastatura i loša (i ružna) kutija. U celini, dobro osmišljen sistem, u „Šutliću" su dobro uradili domaći zadatak (čitaj – nisu ga jednostavno sklepali nego su prvo malo razmislili). Satima je radio kod nas bez ikakvih problema i se veoma malom varijansom u performansama na početku i kraju ciklusa merenja, što je pouzdan znak da je mašina stabilna (bar delimično, ovo je funkcija cale elektronike u jednom čipu).

Ako baš insistirate na AT mašini, ovaj sva kako pogledajte.

Korisne adrese

Šutlić
11000 Beograd, Kosovska 32/1
Tel. 011/332-607
Genelec
75000 Tuzla, Zlokovac 3B
Tel. 075/230-878 lok. 126, 231-557
Fax. 075/237-825, 237 778
11000 Beograd, Jurija Gagrina 115
Tel. 011/179-967, Fax. 011/179-967

Nastavlja se na strani 77

REZAČNICA SLOVA

Na tržištu periferijskih uređaja Roland je poznato ime. Uglavnom ga vezujemo za ono po čemu je najpoznatiji – plotere. Ovoga puta ćemo vam predstaviti uređaj čija osnovna namena nije crtanje raznih tehničkih i drugih crteža nego rezanje slova, ali sasvim dobro može obavljati i tu funkciju. Reč je o uređaju Desktop Sign Marker. CAMM-1, model PNC 1000.

Siniša Mihailović

Nenad Marijanović



Dugačka i ne preterano široka kutija vam govori da je ovo uređaj kod koga papir nije sličan odnosno da se pero kreće po X i papir po Y osi. Kada otvorite kutiju i izvadite PNC 1000, prva stvar koju ćete primetiti ispod LCD displeja je natpis CAMM 1 (Computer Aided Manufacturing Machine). Ovo je uređaj koji objedinjuje funkcije rezaca vinilskih folija i plotera. Dimenzije su mu 735 x 275 x 205mm (dužina x širina x visina). Nalazi se u lepo oblikovanom metalnom kućištu bez boje. Dodu sa metalom i leži na od 13 kilograma otkrivaju solidan rad.

SAM ZA SEBE

U CAMM 1 je ugrađen ventilator, ali je malo nezgodno postavljen – sa donje strane. Ipak nije bilo problema sa grejanjem. Doduše, ne sme se stavljati na mesta sa lošim protokom vazduha kao što su police i slično. Na ovo vam i sam proizvođač skreće pažnju. Zelo što se papir stalno pomera ispred i iza. CAMM 1 treba obezbediti prostor za ovo pomeranje. Kontrolna ploča je prilično jednostavna. Sastoji se od LCD ekrana i 6 membranskih tastera (4 kurzorske, **Enter** i **Function**).

Sa desne strane (posmatrano spreda) nalaze se dve interfejsa. Roland standardno ugrađuje i paralelni i serijski interfejs u svoje uređaje. Na istoj strani se nalazi i prekidač, kućište sa osiguračem i linijski potenciometar za podešavanje visine držača za rezač ili pero. Sa zadnje strane nalazi se ručica pri skivača hartije. Uz CAMM 1 dobijete postolje za papirnu rolnu, 10 držača „alata" (rezača ili pera), metalno „pero" u koje se umeće karbidni rezač i izač, zgodan skalpel za sečenje listova, pincetu, 4 flomastera debljine 0.3 mm i 4 flomastera debljine 2.0 mm, par listova kvalitetnog papira i par folija za testiranje, mrežni kabl, prekidač i solidno uputstvo za rukovanje.

Instalaciona procedura je prilično jednostavna. Ukoliko nameravate da koristite rezač, praktično je sve podešeno, ostaje vam samo da stavite rezač u metalno pero i da pero stavite u držač, spojite CAMM-1 i računar kablom koji nače koristite za povezivanje štampača, uključite CAMM-1 i stavite folju. Zatim pokrenite program koji se dobija na disketi uz CAMM-1, ili neki CAD ili program za crtanje. Znamo da uređaj daje odlične rezultate sa programom *CorelDraw*, ali mi nismo bili u prilici da probamo jer nam nisu isporučeni *Windows* drajveri, pa se svi ispitivanja odnose na ploter kao na softwerski autonoman proizvod. Ukoliko nameravate da povežete CAMM 1 sa računarom preko serijskog interfejsa ili da ga koristite kao plotter, moraćete da izvršite neophodna podešavanja na kontrolnoj ploči.

ZA KONTROLNOM TABLOM

Funkcije birate na kontrolnoj ploči CAMM-1 pritiskom na dugme sa oznakom FUNC. Zatim kurzorskim tasterima birate opciju menija. Kada je ručica pritiskivača hartije (ili folije) podignuta, po uključivanju se prikazuju dva menija između kojih prelazite pritiskom na taster FUNC.

Prvi meni je tzv. konfiguracioni i u njemu postoje opcije za izbor komandnog moda, izbor interfejsa i veličine papira. Postoje dva komandna moda. Prvi se sastoji od 23 tipa komandi, lakši je za programiranje i kompatibilan je sa setom komandi DG DXY plotera.

Drugi komandni mod se sastoji od 56 tipova komandi koje se mogu kombinovati da bi se dobile funkcije visokog nivoa (ekspanzija, kompresija, pomeranje polazne tačke... itd.). Kompatibilan je sa setom komandi RD GL. Sa oba tipa komandi mogu da se odrede mašinske koordinate u rezolucij 0.025 mm. Ukoliko se u konfiguracionom meniju izabere serijski interfejs, automatski se prelazi u meni za određivanje komunikacionog protokola.

Opcijom za izbor veličine papira mogu se birati formati i pravac rezanja ili crtanja: ISO A2, ISO A3, FREE.Y EXPAND.X, ANSI C i ANSI B. Dužina papira ili folije ne mora biti ograničena, ali je sirina ograničena na minimum 381 i maksimum 508 mm. Papir ili folija se pokreću metalnim valjcima, a pritisnuti je gumenim ločkićima. Ima ukupno 5 valjaka za različite dužine papira. Jedan od gumenih ločkića se može pomarati kako bi se pogon prilagodio razlićitim formatima papira ili folije.

Drugi od spomenuta dva menija sadrži opcije za postavljanje parametara izbor alata, brzina sečenja odnosno crtanja, izbor fonta i omogućavanje promena pera za crtanje.

REZAČ ILI PERO

Od alata se može izabrati rezač ili pero označeno brojem 1-8. CAMM 1 nema mogućnost automatske izmene pera. Zato se može omogućiti manuelno menjanje pera po primenju komande za promenu pera ili se komanda može ignorisati. Za svako pero i rezač se posebno odreduje brzina crtanja. Brzina se može odrediti i za rezač. U CAMM 1 su ugrađeni konturni i linijski fontovi. Opcijom FONT birate jedan od 19 setova slova. Izborom rezača automatski se postavlja konturni font. Ukoliko odredite da CAMM 1 ne ignoriše komende za promenu pera, svaki put kada ova komanda bude primljena na LCD displeju cete dobiti obaveštenje da treba da promenite pero uz broj pera koje treba da stavite. Ploter će nastaviti se crtanjem čim pritisnete ENTER na kontrolnoj ploči.

Kada se poluga pritiskivača hartije nalazi u donjem položaju (papir ili folija su umetnuti) pritiskom na taster FUNC možete birati između još 5 menija. Ukoliko nemate računar ili vam je računar trenutno zaposlen, ne znači da ne možete upotrebljavati rezač. Kada se folija nalazi u CAMM-1 na kontrolnoj ploči pritiskom na taster FUNC dodijale meni u kome podešavale parametre za manuelno rezanje. Izborom prve opcije menija možete upisati tekst koji želite da bude izrezan. Slova se unose kurzorskim tasterima. Maksimalna duzina teksta je 20 znakova. Sledećom opcijom menija možete podesiti sirinu znaka i činu celog teksta. Treća opcija služi za izdavanje komende rezaču za početak rezanja.

PAPIR ILI FOLIJA

Sledeći meni služi za pomeranja papira ili folije. Ne raspolaganju su vam opcije za pomeranje papira ili folije za punu sirinu za jedan deo kako biste videli sta je uradeno na početnu poziciju ili do kraja površine za rezanje (crtanje) i nazad.

Meni REPLOT vam omogucava ponovno šlampanje (rezanje) prema podacima uskladištenim u REPLOT beleru. Podaci se u REPLOT beler šalju iz računara na zahtev – izborom opcije menija.

U toku štampanja ili rezanja možete napraviti pauzu pritiskom na taster ENTER. Tada vam se pružaju mogućnosi nastavka posla, pražnjenja I/O belera ili izbora ostalih menija. Zadnji meni sadrži pari različitih opcija za verifikovanje postavljenih parametara, promenu jedinica za prikaz koordinata i podešavanje/spuštanje alata.

Svi meniji su dodio objasnjeni i ilustrovani u uputstvu za upotrebu. Posebna pažnja je posvećena izboru formele papira i pravaca rezanja (crtanja). U uputstvu za upotrebu su dale obe seta komandi uz dodro uradene primere njihove upotrebe u bejziku.

Iako CAMM 1 nije nov proizvod, mogućnosti koje donosi i dalje odišu svežinom. Izradi se malo šta može zameniti. Mogu se jedino dati predlozi, možda bi bilo dobro ventilator premestiti sa strane. Primerljivo zgleda ideja da se uz CAMM 1 isporučuje postolje kao kod velikih Rolandovih plotera. U tom slučaju ventilator je na idealnom mestu. Za ono kome višebojni ploter nije neophodan, CAMM 1 je idealno rešenje. Mogucnosi rezanja na samolepljivim folijama, pri čemu je veličina ograničena samo činjenicom od 500 mm i fizičkom dužinom folije, otvara nova područja primene.

Tabela 1 Roland CAMM-1: tehnički podaci

Dimenzije papira	381 – 500 mm ISO A2, ISO A3, ANSI C, ANSI B
Pomeranje papira	ISO A2, ISO A3, ANSI C, ANSI B, FREE.Y, EXPAND-X
Orudja	Rezač (karbid), pera
Brzina rezanja	20 – 150 mm/sec u koracima po 10, u svim pravcima
Sila na peru	50 – 100 g
Rezolucija mehanizma	0.05
Softverska rezolucija	0.025
Preciznost	± 0.2% ili ± 0.1 mm
Interfejsi	Paralelni, serijski
Uvlačenje listova	Mehanizam sa valjcima
Veličina belera	1Kb (8Kb REPLOT beler)

Preduzece MZ
11000 Beograd, Dubljanska 70
Tel. 011/434-812, Fax. 011/450-471

ŽIVE SLIKE

Do sada u svetu DOS personalnih računara nije bilo nijednog ozbiljnog programskog paketa za fotorealističnu 3D animaciju u realnom vremenu. Razlozi za to su bili relativno slaba procesna snaga personalaca, MS DOS koji je sa svojom 640 K barijerom u startu blokirao svaku pomisao na ovakav program, kao i slaba rezolucija, broj boja i brzina grafičkih kartica.

Nebojša Novaković

Autodeskov *3D Studio* je prvi ozbiljan paket za fotorealističnu 3D animaciju profesionalnog kvaliteta za 32 bitne DOS personalce. *3D Studio* se pojavio zajedno sa 486 masinama koje predstavljaju minimum za bilo kakav ozbiljniji 3D grafički rad. Najveći deo problema malorog MS DOS je resen pomocu 32-bitnih DOS Extendera kojima se otvaraju vrata neogranicenih kapaciteta zaštićenog načina trideset2 32-bitnika. Na PC tržištu su sada deseline vrsta različitih mocnih grafičkih kartica u lepezi rezolucija i performansi. Sve to zajedno znaci da je nestao najveci deo prepreka koje su stajale pred ovom primenom personalnih racunara. Po svom naslanku *3D Studio* neodoljivo podseca na *AutoCAD*. *AutoCAD* je kada je nestao bio prvi ozbiljan a ne i preferano skup CAD paket za personalce. *3D Studio* je prvi ozbiljan a ne i preferano skup PC paket za 3D animaciju. Naravno ne treba izgubiti iz vida da oba ova proizvoda nose znak Autodeska.

Trodimenzionalna animacija je najbitnija komponenta multimedijalne primene racunara. U saradnji sa digitalizovanom slikom i zvukom kao i odgovarajucim dodatnim hardverom mogucnosti za primenu su neograničene.

NA PRVI POGLED

3D Studio dolazi u ekstravagantnoj crnoj kutiji veličine oko 30 x 23 x 10 cm. Kutija sadrži 3 priručnika i instalacione diskete. U dokumentacioni glavni deo predstavlja *Reference Manual* sa oko strana koji je veoma dobro uraden – bolje i od *AutoCAD 11 Reference Manuala*. Druge dve manje knjige su *Tutorial* za korisnike bez iskustava u ovoj oblasti (a takvih je daleko najviše) i vodič za instalaciju i konfigurisanje.

Program je smesten na 7 instalacionih disketa. Instalacija se vrši isto kao i kod *Auto CAD 11* – startovanjem rutine INSTALL sa diska 1. INSTALL je delo Knowledge Dynamics Corp. iste kuće koja je napravila INSTALL za ostale Autodeskove proizvode. Ukoliko imate Weitekov koprocesor, sivan će raditi nešto brže, ali prilikom instalacije moraće izabrati Weitek verziju. Naknadna kupovina Weitekovog 3167 ili 4167 traži ponovnu instalaciju programa. Posle desetak minuta instaliranja program je na tvrdom disku.

Kada u osnovnom 3DS direktorijumu otkucamo 3DS, posle određenog vremena potrebnog za inicijalizaciju (dosta zavisnog od brzine računara) pojavljuje se osnovni ekran *3D Studia*. Meniji sa desne strane nešto širi od onih u *AutoCAD*u gornji red sa statusnim podacima i par padajućih menija kao i donja reda za unos podataka o elementima. Za razliku od *AutoCAD*. ovde se padajući meniji i meniji sa desne strane dopunjavaju. Meniji sa desne strane izvedeni su u hijerarhijskom sistemu s tim što je i ovde vizuelna predstava drugačija od *AutoCAD*. Dok u *AutoCAD*u podmeni na desnoj strani zamenjuje prethodni meni u *3D Studiu* podmeni se pojavljuje dva karaktera uvučeno ispod zadnje opcije glavnog menija bez obzira na to koja je opcija glavnog menija bila izabrana. Uz pomoc dobro izabranih boja za oznacavanje izbranih opcija željene operacije se biraju veoma vrzo. Ispod ovih menija nalazi se nekoliko grafičkih ikonica za neke osnovne ekranske operacije. Padajuci meniji u gornjem levom uglu sadrže naredbe za davanje podataka o verziji softvera, učitavanju modela, konverziju *AutoCAD* DXF i FLM datoteka itd. A u gornjem desnom uglu ekrana piše *3D Editor*. Šta je *3D Editor*?

OSNOVNI MODULI

3D Studio se sastoji iz nekoliko modula, koji su medusobno usko vezani i koji u biti podsecaju na fabričku traku sa pojedinim elepema proizvodnje u posebnim elapama. Tako se i put od zamisli modela do realizacije, bilma o njemu u *3D Studiu* odvija u više elapa – modula.

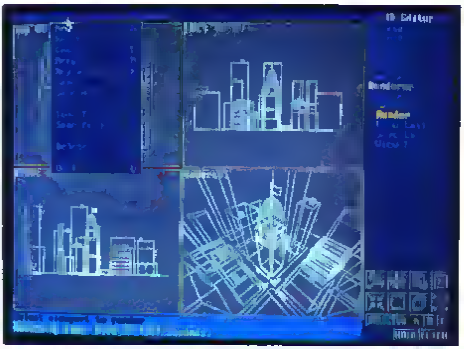
3D Editor je srce paketa *3D Studio*. U njemu se kreiraju svi elementi modela, generišu složeniji objekti iz prostijih, postavlja scena za senčenje na način analogan onome kod fotografa – kemere i svetla – i dodeljuju materijali, elementima. Iako da pri senčenju oni dodijaju realističnu predstavu nalik onoj u prirodi. *3D Studio* podržava više vrsta senčenja, prosti zidani model, zatim *flat* senčenje gde svaka površina ima zasebnu nijansu u zavisnosti od izvora svetlosti, *gouraud* senčenje gde nijansa ose svake površine određuje interpolirane vrednosti između nje i druge površine i najsponji ali i najrealisticniji *phong* način senčenja gde se nijansa izracunava za svaku tacku. *Phong* senčenje omogucava i generisanje bacenih senki jednog objekta na drugi. *Ray Tracing* ili pracenje zraka kao vrhunski domet generisanja fotorealističnih prikaza, nije za sada podrzan u *3D Studiu* – podržava ga na primer konkurentski paket *Big D*. Inače izbor 3D tela koja se mogu neposredno generisati približno je jednak sposobnostima *AutoCAD* 10. Ako ortaz unosite iz *AutoCAD* putem DXF ili FLM datoteke, *3D Editor* je za vas prvi korak.

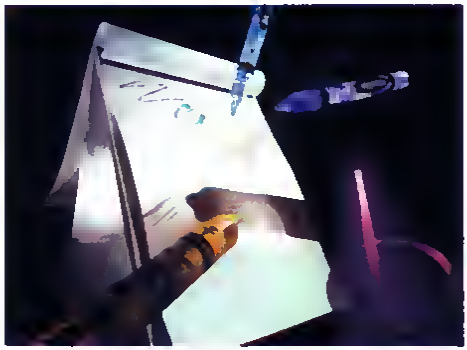
Za one koji model od početka prave u *3D Studiu* a model ima puno različitih zakrivljenih i drugih specifičnih površina prvi korak je *2D Shaper*. *2D Shaper* se koristi za generisanje različitih dvodimenzionalnih krivih koje se u drugim modulima koriste za dobijanje 3D elemenata. U *2D Shaper*u se generišu dve vrste krivih: prve koje su preseci objekte i druge koje su putanje za izvlacenje' krive u prostor. U ovom modulu se naravno mogu praviti i ostale 2D linije i povrsi (pravougaonici, krugovi, elipse itd.) koji pored napred navedenih funkcija mogu biti neposredno preneseni u *3D Editor*.

3D Lofter je modul u kojem se od krivih iz *2D Shapera* generišu trodimenzionalne površi s tim što putanja za izvlacenje krive u prostor sada može biti i prostorna kriva. Na taj nacin se mogu dobiti prakticno bilo kakve površi. *3D Lofter* kao druga etapa generisanja modela je prvi od modula sa prostornim koordinatnim sistemom i 4 prozora na ekranu. Tri manja na levoj strani su tri osnovne projekcije modela a četvrti je izometrijski ili (u *3D Editor* u i *Keyframer* u) i perspektivni pogled.

Materials Editor je modul za kreiranje i izmenu materijala koji se dodeljuju površinama modela. Svaki od materijala ima svoja svojstva kao boju, sjaj, providnost i neku teksturu (na primer zid od cigala). Ova svojstva se u *Materials Editor* u lako i brzo menjaju pomocu regulatora i izbore vec gotovih bit map tekstura (PCX crteži takođe mogu poslužiti). U osnov







noj biblioteci *3D Studia* dostupno je 105 različitih izvrsno predstavljenih materijala, bo bele plastike, preko raznih vrsta drveta i zleta do ornog stakla. Svaki bo njih se može isebrati i proizvoljno menjati u *Materials Editor*-u, pri čemu se izmene interaktivno vide u nekom od pet malih prozora namenjenih tome. Novobodbijenim materijalima se dbodeljuju druge imene, teko da možete dopunjavati loneko prilično bogatu biblioteku. *Materials Editor* je, ineče, jedini bo mbaule koji stalno radi u rezoluciji 320×200 i porazno liči na okolinu u *Animator*u.

Poslednji mbaul je *Keyframer*. Namenjen je pripremi i generisanju animacije. Položaj modele iz *3D Editor*-a, sa svim kamerama i svetlima je ujedno prva slika u *Keyframer*-u. Mogućnosti ovog mbaula su zaista fantastične u animaciji se mogu pomerati objekti, kamere i svetle po nezevisnim putanjama, objekti mogu biti skeletni u bilo kojoj dimenziji (smanjivani, povišavani sužavani...), mogu biti i deformisani ili možete jednostavno zadati dve različite slike sa različitim položajima objekata ili kamera i zadati programu da sam napravi animaciju sa proizvoljan broj slika između. Potom se vrši generisanje pojedinačnih slika i njihovo vezivanje u jedinstveni fajl.

OŽIVLJENE SLIKE

3D Studio, kao i *AutoShade* raspolaže mogućnošću sa bavojene ekrane, jeden se grafičkim editorom za editovanje i neredbe, i drugi namenjen prikazu renderovanih slika (senčeno predstavljenih modela se meterijalizacijom i svetlosnim izvorima). Sveki bo tih ekrana more imati poseban grafički adapter. Po Autodesku, najčešće solucije je VGA za editorski ekran i VIŠTA III TARGA za rendering ekran. Medutim, neme sveko mogućnost za dve ekrane. U tom slučaju i editovanje i rendering su na istom grafičkom adapteru i ekranu. Kao što znamo, broj istovremeno dostupnih boje na ekranu neposredno zavisi bo broja bitnih ravni u video memoriji grafičke kartice. Današnje grafičke kartice imaju izmedu 4 i 32 bitne ravni. Pošto se broj dostupnih boje izračunave kao 2 na n-ti stepen gde je n broj bitnih ravni, znači da današnje grafičke kartice teoretski mogu imati izmedu 16 i 4 3 milijarde istovremeno dostupnih boje. Naravno, maksimum koji današnje kolor palete mogu pružiti je „samo" 16 8 miliona (2 na 24) njiansi – 24 bita za boju. Ostalih 8 bitnih ravni se može iskoristiti za preklapajuće, kursorske ili teket ravni nezavisna bo seme slike.

Autodesk *3D Studio* eve renderinge interno izračunave na 24 bita. Nekon izračunavenja evih ekreneklih tačaka tih 24 bita se ukoliko kartica neme toliko bitnih ravni interpolira na 8 ili 16 bita i slična je gotova. Ako boeberemo rendering na diek fajl, dobičemo dve otveri veću brzinu, jer program ne mail na ekran, i mogućnost slobodnog izodre rezolucije i broja boje. Sve do 8192 x 4096 u svih 16 miliona boje. Takve slike koje su u standardnom GIF ili TIFF formatu možete, na primer, posleti direktno na slajd rekorder. Ovakve mogućnost je idealna za prezentacione svrhe. Naravno, niko neće ni pomisliti de ovakvu mogućnost koristi za generisanje animiranih sekvenci, iz razumljivih brzinskih razloga.

Brzina renderinga u *3D Studiu* neposredno zavisi bo nekoliko faktora. To su:

1 – brzina CPU i koprocesora,
2 – kapacitet i brzina RAM,
3 – brzina dieke, ukoliko kapacitet RAM nije dovoljan
4 – razolucije renderinga,
5 – rendering na ekran ili diek,
6 – vrsta senčenje,
7 – primene antialiasinga (omekšavanje nazubljenih vektora),
8 – ono najvažnije, složenost modele.

Ukoliko imate dovoljno brz računar (50 MHz 486, na primer) elobode izbora načina renderinga će biti znatno veće. Naravno da jeden 386SX se 4 Mb RAM nije pogoden za red se mbaslima većim bo 1 Mb i rezolucijema većim od 640×480.

PRVI UTISCI

Prva što se kod *3D Studia* primećuje je de su ekranske operacije brže, i to znatno brže, nego u *AutoCAD*. Očigledno je de je pri razvoju ovog paketa prednost imele neke drugačije logike rede otveri nego one u *AutoCAD*-u. S druge strene, nekome ko se saživeo se *AutoCAD*-om, njegovim korisničkim interfejsom i logikom rada *3D Studio* neće biti baš po voji. Pošto takvih ima dosta, i se ovim programom bi mogla de se deel ono što se deslio se *AutoSolid* paketom, koji je imeo izvenreden korisnički interfejs, daleko bolji od *AutoCAD*-a, ali je nestao, integrišući se u *AutoCAD* 11 kao AME (Advanced Modelling Extension). S druga strane, *3D Studio* je namenjen i se umetničke poslove, dakle i onima koji *AutoCAD* možda nikada nisu ni videli i na seviee bo njegovog korisničkog interfejse.

Upravljanje osnovnim ekranskim operacijeme (**Redraw Zoom Pen** itd.) preko ikonica nije loše ideje, a tim što to već odavno postoji i u *AutoCAD*-u kod displej-list drejvere za kartice se grafičkim procesorima.

Lakoća s kojom se u *3D Studiu* dobijaju praktično bilo kakve tele je zaiste za poštovenje. Jedine dve otveri koje se mogu zameriti su nedostatak logičkih operacija na telima (kao kod AME solid mbodelere) i nemogućnost generisanje složenih NURBS (Non Uniform Rational Bezier Spline) krivih.

Mogućnost neposrednog renderinga na diek u GIF, TGA ili TIFF formatu, bez obzire koje je rezolucija grafičke kartice, omogućuje generisanje izvenrednih slike za prezentacije i druge potrabe koje se bez problema mogu prenositi u druge programe, štampati na kolor štampačima ili slati na slajd rekordere. Animacije se, pak, mogu snimati u Animatorovom FLI formatu, tako de se mogu posmetreti i bez 3D Studia.

3D Studio, ne želoet ime i par problema. Solid modeli iz *AutoCAD* 11 AME ne mogu se prenositi dok se ne prebace u mrežni prikaz a i onda ih u mnogim slučajevima program ne prime. Prilikom unosa DXF modele nekoliko puta se dešavalo da *Keyframer* „zaboravi" da neki delovi mbaele postoje i jednostavno ih preskoči. Na zadovaljno lpak, da je ovo verzije 1.0, i da je eveki početak težak.

NIŠTA BEZ HARDVERA

3D Studio se konfiguriše na drugačiji način nego *AutoCAD*. Srce svega je 3DS.SET fajl. U njemu korenik zadaje direktorije za rezlička datoteke definiše memorijske adrese koje će program koristiti i putanje drejvere za grafičke kartice, printere i upravljače video rekordere. 3DS.SET je tekstualne datoteke i bez problema se edituje u bilo kom ascii editoru.

Najveći problem su, svakako, grafičke kartice. *3D Studio* ime ugrađenu podršku samo za standardnu IBM VGA. Truevision VISTA i TARGA kartice se video izlazom. Truevision kertice imaju rezolucije 512 x 486 i 752 x 486 u do 16 8 miliona boje istovremeno dostupnih. Medutim, ove kertice, kao i e njime kompatibilne (Matrox Illuminator i Everex Vision 18) su relativno skupe i neme ih baš mnogo. VGA je podržane se 640 x 480 u 16 boje ze grafički editor i 320 x 200 u 256 boje za rendering. To je, naravno, nedovoljno ze slike koje bi ličile na nešto. Medutim, ukoliko vaše kertica ime ADI P386 drejver za *AutoCAD* 11 i *AutoShade*, *3D Studio* će moći da iskoristi njene mogućnosti.

Veliki problem kod *3D Studia* je što je, berem za sade, nemoguće preuemaravenje be-



rem dela grafičkih operacija sa CPU ne neki grafički procesor. Sav posao je na procesoru u PC i direktno zavisi od njegova brzina. U toku je razvoj posebnih postprocesora za rendering i animaciju koji bi taj posao sellli na recimo, Intelov 860, dok se na pojavi potpuna 860 verzija *3D Studia*.

3D Studio ima malkice strožije hardverske zahteve nego *AutoCAD* 11 386. Traži se minimalno 386SX sa koprocesorom, 3 Mb RAM, i dasetak slobodnih megabajta na disku posle instalacija. Optimalna konfiguracije trenutno je 33 ili 50 MHz 486 sa Weitek 4167 koprocesorom, 16 Mb RAM, 200 MB ili većim HD i grafičkom karticom opsežnom da de beram 32.768 boja istovremeno na rezoluciji rada 640 x 480 (to mogu biti i ET4000 SuperVGA). Trenutno grafički procesori nisu bitni, jer ih ovaj program za sada ne može iskoristiti. Portiranje na 860 će doneti ubrzanja i od više dasetina puta u generisanju fotorealističnih slika. Dolazak 586 sa svojim FP koprocesorom višestruko bržim od onog u 486 (ali i višestruko sporijim od onog u 860XP) će, s druge strane eliminisati Weitekove koprocesore iz buduće trke.

Autodesk još nija predstavio verzije *3D Studia* za druge platforme, kao recimo Macintosh III Sperostation mada su ona u planu. Nesumnjivo je da će na ovakvim računarima pod odgovarajuće kapacitetnijim operativnim sistemima i sam paket biti brži i imeti lepši korisnički interfejs.

3D Studio je planir u svetu personelaca. Prvi paket s kojim i ne „običnim" 386 ili 486 PC možamo stvarati (i s uživanjem poemetrati) svoje fotorealistične filmove nalik na recimo TRON. Već se pojavio i odgovarajući dopunski paket *Record Studia* za digitalno nasnimavanje zvuka za *3D Studio* filmova ne video traku zajedno sa filmom nerevno. Na putu su i druge dodatne aplikacija koje će *3D Studia* zaista učiniti pravim multimedije pekatom, mnogo pre no što to postane *Windows 3.1*. Opreme koju ovo zahteva, počevši od snažnog računara, preko VTR kontrolere profesionalnog video rekordera i još par sitnica. slične jeftinoće udaljava ovaj proizvod iz sfera hakerskih interesovanja. Međutim onome kome je ovo potrebno brzo će se isplatiti.

Korisno adreso

OSA računarski inženjering
11130 Beograd, Miće Stojkovića 6
Tel: 011/557-592, Fax: 559-845

PEDIGRE I BRZINA

O firmi „Nth Graphica" amo već pisali na stranicama „Računare". Sada nam se pruža prilika da pogledamo njihov novi proizvod namanjan „narodnim masama" pod jednostavnim nazivom „model 150".

Dejan V. Veselinović

Sema kartica ima puni AT format, kako po dužini tako i po visini. Nija pretrpane elektronikom, mada nije ni prazne: tu su četiri čipa u SMT tehnologiji, jedan VLSI (Bt 468KG110 RAMDAC) u keramičkom pekovenju i sa postoljem, 9 PAL čipova, 17 TTL čipova i 16 memorijskih čipova (TC 524256AZ-10) u ZIP pekovenju, takoda sa postoljima. Pet oscilatora i jeden Actel VLSI čip kompletiraju eliku. Koliko emo mogli da nazremo, štampe je (bar) četvoroslojne i veome uredno uradene; utisak kvare dve žice koje sa vode preko maltene pole ploče, pe kroz nju do konaktore ne poleđini. To, kao i oznaka za diode kojih nema, jer su zamenjene kratkospojnicima, daje utisak de ovo još uvak nija itamcije zrele faza, odnosno de je ovo proizvod ne koma će se još raditi.

Dekle, prezantecije ploče je vrlo dobre, iako je cane odlične. No dokumantacija za iste odlične para je i više nego što ste pietili, jednostavno superlativne, na samom vrhu klase u koju same kerta spede. Odlično je eredana, odlično prezentirana, sa indaksom i jasnim uputstvima kade šta urediti i kako. leto važi i ze dopuneka usluga koja se opciono nude uz karticu, ima evege, košta kao evetog Petra kaljena, eli radi bez greška. Uprkoe pominjanju narodne klase, firme „Nth Engine" je naiik Mercedes-Banzu-u, i mali model je skup, ali zato ime debeo pedigre. Tu je i veznik se MS Mouse Bue miša, da se nade.

Karta je namenjene radu u sistemime se VGA kartom, ali i sistemima bez nje. Naime, standerdna opreme uključuje i jeden propuseni („pese-through") kabl se dve vrste veznike za VGA kartu (klasični pozlećeni konektor ili igličasti veznik), čima se omogućeve povezanja dva monitore istovremeno povazane za sistem. Ako vam je ovo jedina kartice, obredovaća vam da znete de će se one eame prilagoditi nekoliko postojećih standardnih režime neravno, pomad sopstvenih režime. Ako na radi u evom najboljem režimu sa razluciiom od 1280×1024 i 256 od 16,7 milione boje (naravno bez preplitanja), sama će pokušeti sledeći korak nadole - e to je 1024×768 se ili bez preplitanje, e ako ni to na uspe (recimo, program nema takvu pobudu), epustiće se na bednih 640×480. Nikeko ne možete da izgubite.

Proizvodi firme „Nth Graphics" su u glavnom orijentisani ka visoko profesioneinim programima, e u okviru njih, pre svega ke dosajenu u toj oblasti, firmi Autodesk. Prema tome karta podrževe program *AutoCAD* verzije 10 i 11, *AutoShade*, *AutoSketch* i *3D Studio*, te RenderMen. Porad ovoge, karta podrževa i *MS Windows 3.0*, e sade sa pojevila i pobuda ze najvišu rezoluciju (1280×1024), podržani su i *X Windows* odneono UNIX (u vidu kompatibilnosti sa IBM 8514/A). Uopšte karta podrževa eve programe koji imeju pobudu za IBM 8514/A.

Nakon uspešne instelecije (što zapreve nije ništa posebno teško ili komplikovano), probali smo kartu u prolaznoj verziji („pass-through") i samostalnom radu. De radi, bogami radi baš onako kako kažu da treba, ali, eli ...

Odmah de kažemo: nikakvih neočekivenih problema nismo imali, bar ne onakvih se koje ne kraju nismo utvrdili sopstvenu greškU. Visoka rezolucija, sve iznad VGA, lepo radi e ono „all" se odnosi ne brzinu rada. Karta je brza, nema spora, vidno odnosno osetno brže od naše inače prilično brze Tseng ET4000 karte, ali nedovoljno brze u odnosu na cenu. Mede jesmo obavili kiasična merenja, od svih testova koje inače koristimo mogli smo da upotrebimo samo jeden, e i njege samo delimično, pa zato nema grafikone i neme tabele brzine rade. Ono rezultate koja emo dobili pokazuje da je karta bezmalo duplo brža.

To neravno uopšte nije zanemariv rezultet kade se zna de se poradi sa brzom kartom, ono što odekaćе je cena ovog proizvoda u odnosu na konkurenciju, kao uostalom i ceo paket. Ova karta u odnosu na standardne kerte, nudi 33,3% veću rezoluciju i oko 60% veću brzinu, ali za (prama reklamnim cenama) oko desat pute veću cenu od kartica kojime konkuriše (u oba slučaja, uzete su cene u inostranstvu, preme prospektime proizvodača ze najjeće modele).

Iz dokumantacija je jasno da je ova karta namenjene pre svege veome jakim i zahtevnim korisnicima programa firme AutoDesk, e pre svege korisnicime programe AutoCAD. Za ove programe postoji posebne pobuda, koju niemo probali jer ne poseduiemo i ne poznajemo ovaj program, no, verujemo da se „Nth Graphics" potrudio, e ne osnovu svega ostalog, a pre svega lakuetve sa prethodnim slučajem, testirenjem kartice se transpjuterima, ni najmanja ne sumnjemo de su posao obavili udemički. Za ovekve korisnika, ova karta bi mogla biti odlično rešenje, kao i se druge korienike AutoDesk-ovih programe.

Druga grupa korisnika bi mogli biti evi oni koji koriste razne programe pod *Windows* okružanjem, kao recimo *Ventura*, *Corel Drewi*, *Aldus PageMaker*, rasni CAD programi...

Za sve ostale ovo je proizvod koji nema svog emiela ueled valike cene i relativno male dobiti u poredenju sa „normalnim" super VGA kartama, koje su joj jednake po broju boje i jedve 33,3% ispod nje po rezoluciji.

Dakle, neobičan i skup proizvod, sa poznatim imenom i jasnim pedigreom koji povlače i visoku cenu, namenjen zeleta profesionalnom segmentu tržišta, ali ako spadate u taj segment preporučili bismo vam de obavezno pogledate ovaj proizvod. Naprevljen je za vas.

Korisna adresa

ASV Graphics
11000 Beograd, Andrićev venec 4
Tel/Fax: 011/345-277

IGRE SA SLIKOM

Danas postoji već veliki broj programa za grafičku prezentaciju podataka. Iako se međusobno razlikuju po izgledu i načinu rada, svi imaju zajednički cilj: prikaz informacija putem slike. Zbog velikog obima i kvaliteta ponude i cenovnog raspona, teško je dati odgovor na pitanje „koji izabrati". Jedno je sigurno: nemojte potceniti naoko sitne proizvode.

Siniša Mikajlović

Izborom programa za prezentaciju posao najčešće nije završan. Treba odabrati i „alate" koje ćete koristiti uz taj program. Pod „alatima" se najčešće podrazumevaju programi za crtanje, skeniranje, konverziju iz jednog u drugi format i ostale operacije nad slikom. Cilj ovog teksta je da vam prikaže dva programa korporacije AimTech: *IconAnimate* – program za animaciju i *RezSolution* – program za konverziju i prikaz slika u PCX formatu.

SAMO POD PROZORIMA

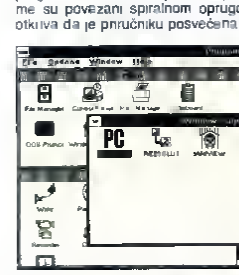
Do pre samo nekoliko godina grafički korisnički interfejs na PC kompatibilnim računarima bio je prava retkost, a mišljenje o potrebama za njim krajnje podeljena. Onda je Microsoft na svetlo dana izneo svoj *Windows* koji je doživeo veći uspeh nego svi ostali programi te namene zajedno. Prirodno je da jedan uspešan softverski proizvod dobije veliku podršku. U ovu podršku spadaju i dva navedena programa. *IconAnimate* i *RezSolution* rade samo uz *Windows*. Onaj deo računarske populacije koji ne koristi *Windows* će verovatno ovo shvatiti i kao najveću manu ovih programa.

Oba paketa se dobijaju u lepo dizajniranim kartonskim kutijama. U paketima se nalaze po dve diskete (jedna 5.25" i jedna 3.5") uputstvo za instalaciju, uputstvo za upotrebu i anketni listić. Instalacija se obavlja pomoću priloženog programa. Priručnici koji se dobijaju uz programe su povezani spiralnom opругom. Sadržaj otkriva da je priručniku posvećena sva potreb na pažnja. Sve opcije se detaljno objašnjene, pregledno izložene i ilustrovane. Na kraju priručnika se nalazi kratak indeks. Priručnik za *IconAnimate* ima 178, a za *RezSolution* 54 strane. Programi i priručnici na oba paketa nose oznaku verzije 3.0.

ANIMACIJOM DO PREZENTACIJE

IconAnimate omogućava prikaz i animaciju bilo koje slike u *.PCX ili *.WMF formatu u bilo kojoj rezoluciji koju podržava *Windows*. Za razliku od većine sličnih programa, kod kojih se animacija sastoji od sukcesivnog prikazivanja određenog broja slika, *IconAnimate* omogućava animaciju pomeranjem delova slike što znači da pozadina može da bude statična dok se deo slike pomera ispred ili iza nje.

Da bismo kreirali animaciju objekti tj. delovi slike i slike koje predstavljaju pozadinu ili sekvence animacije moraju biti iscrtani i snimljeni kao posebne slike. Crtanje ćete naravno obaviti u nekom programu za crtanje. *IconAnimate* ne poseduje mogućnosti crtanja kao na primer *Autodesk Animator*. *IconAnimate* je tzv. *script-oriented* program što znači da ceo proces animacije mora biti opisan i snimljen. To opisivanje se vrši izodrom određenih ikona iz **toolbox** prozora. Svake od ikone predstavlja neku od operacija kao što su promena boje, prikaz pozadine ili dela slike, animacija određenog dela slike, efekat itd.



Instalirane ikone za *IconAnimate* i *RezSolution* u desktopu

Proces pomeranja i kopiranja ikona u skriptu izgleda kao u većini leksi editora. Iza svake ikone koja predstavlja deo skripta može se dodati tekstualni komentar. Program omogućuje osnovna operacije sa blokovima premeštanje, kopiranje i slično. Moguće je reprodukovati animaciju na obeleženom delu skripta – bloku. Posloji širok izodr ela kela kao što su zaismnjivanje preklapanje postepeno bri sanje zumiranje osvetljavanje pro zvoljne oblasti slike (flash).

Uz *IconAnimate* se dobija i program *IconAnimate View* za reprodukciju animacija napravljene programom *IconAnimate*. Naravno reprodukcija je moguća i iz osnovnog programa *IconAnimate* je solidno urađen program. Jedina zamerka je ovisnost o *Windows*-ima što ga čini praktično neupotrebljivim na sporijim mašinama.

IZ REZOLUCIJE U REZOLUCIJU

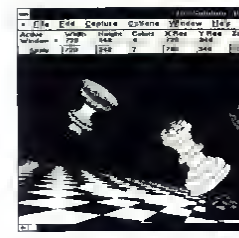
RezSolution je program za konverziju slika u *.PCX formatu. Verovatno pogađate na osnovu imena, da se ne radi o konverziji iz jednog u drugi grafički format, već o konverziji iz jedne rezolucije u drugu u okviru istog formata. Možda ovo nekome zvuči čudno – kakva je konverzija potrebna kada je pod *Windows*-ima moguće prikazati sliku u bilo kojoj rezoluciji? Razlika je u distorziji odnosno izobličenju slike. Na primer, slika kreirana u rezoluciji 640 X 200 će delovati spljošteno kada se prikaže u rezoluciji 640 X 480. Ovaj problem rešava *RezSolution* tako što automatski menja dimenzije slike da bi je prilagodio drugoj rezoluciji. Dimenzije slike možete menjati i po želji. Ovo je poznatije kao zum. Tako zumiranu sliku, ili deo slike, možete i snimiti.

RezSolution omogućava i konverziju odja u njanse sivog. Ovo je veoma korisno ukoliko se bavite DTP-om ili ukoliko vaša prezentacija radi na monohromatskom, a pravljena je na kolor sistemu. Posloji i mogućnosi promene odja ne slici. Na primer, ukoliko ste snimili dvoodnu sliku možete je „promeniti" u 16-bojnu. Ovo menjanje ne znači da će se pojaviti 16 boja, već da će se u *paint* programu pojaviti paleta od 16 boja, pa će odjenje biti moguće. Sam posao odjenja ćete (kad i obično) moreti sami da obavite.

Veliki broj programa za pregled slika u *.PCX formatu pri pregledu slika u 256 odja koristi aktivnu paletu sa 256 odja. Kao posledica ovog može vam se desiti da dobijate neku psihodeličnu kombinaciju boja. Da biste izbegli slične egzotične kombinacije boja *RezSolution* snima celu paletu zajedno sa slikom.

Da li vam je zatrebalo da odštampate neki ekran iz *Windowsa*? Neki programi koji „hvataju" grafički ekran iz DOS-a jednostavno odbijaju da rade pod *Windowsima*. Primer je *WordPerfectov Grab*. U *RezSolution* je ugrađena i ova opcija. Možete „uhvatiti" sliku proizvoljnog dela ekrana, prozora ili čitavog ekrana.

Kao i kod *IconAnimate*-a jedina zamerka ovom programu je ovisnost o *Windows*-ima. Korisnici koji ne koriste *Windows* bi verovatno pozdravili mogućnosti startovanja programa iz DOS-a u nekoj od budućih verzija. Ovakvih alata nikada nije dosta. Dobrodošao dodatak bi bila opcija za konverziju između različitih grafičkih formata. Time bi ovaj program mogao na jednom mestu da pruži većinu usluga potrebnih pri manipolaciji slikom.



Animacija uhvaćena programom *RezSolution*

Korisna adresa

AimTech Corporation
20 Trafalgar Square
Nashua NH 03063-1973
Tel (603) 883-0220 (800) 289-2884
Fax (603) 883 5562

KRALJ U NOVOM RUHU

U odnosu na ranije verzije Majkrosoftovog makroasemblera, verzija 6.0 predstavlja pravi kvalitativan skok napred, čime je i opravdan skok u numeraciji (poslednja verzija je bila 5.1). Program se isporučuje u formi koja sve više podseća na pakete viših programskih jezika (posebno C), uz solidan *CodeView* dibager i nezaobilazni PWB (Programmer's WorkBench). Nizom novih komandi za kontrolu programskog toka, mašinski jezik se približava C-u ili paskalu, mireći ključne prednosti „viših" i „nižih" programskih jezika -efikasnost u programiranju i efikasnost u izvršavanju.

Jovan Skuljan

Macro Assembler 6.0 isporučuje se na četiri standardne AT diskete (od po 1.2 MB), označene kao *Setup*, *DOS Utilities*, *OS/2 Utilities* i *Microsoft Advisor*. Na *Setup* disketi nalazi se, između ostalog, i istoimeni program za instalaciju softvera. Interesantno je da se većina dataloteka dobija u komprimovanom obliku, čime se dosta štedi prostor na disketama. Na taj način uloga programa *Setup* nije samo da prekopira fajlove na hard disk, već i da ih uspuli raspakuje. Osim toga, program nam u uvodnom manju nudi još nekoliko opcija, kao što je kopiranje (i raspakovanje) po volji izabranih datoteka, pregledanje informativnih fajlova README.DOC i PACKING.LST i slično.

Instalacija asemblera je jednostavna i relativno brza. Konsnik mora da odgovori na nekoliko pitanja koja se liče operativnog sistema i konfiguracija, da propusti sva diskete kroz flopi drajv i, na kraju, da modifikuje AUTOEXEC.BAT i CONFIG.SYS po prilozenom uputstvu.

Za pohvalu je opcija takozvane *default* instalacije, gde nam program sam nudi listu uobičajenih odgovora na sva pitanja. Dovoljno je pregledati listu, dati svoju saglasnost i proces instalacije će teći automatski.

Na diskelama se nalazi preko 90 fajlova koje program *Setup* kopira na hard disk. Taj posao traje nekoliko minuta, što je obično dosadno čekati. Međutim, Majkrosoft se dosetio da zabavi korisnika time što u toku instalacije, osim uobičajenih poruka za zamenu disketa, sve vreme sloji ispisane informacija koliko je datoteka još preostalo za kopiranje.

Macro Assembler 6.0 može se instalirati za DOS ili OS/2 po želji. Mi smo isprobali DOS verziju na jednoj 386-ici i na osnovu dosadašnjih utisaka slobodno možemo reći da je to pravi kvalitativan korak napred u odnosu na dosadašnje verzije programa MASM.

Asembler sa svim pratećim programima i datotekama zauzeće na disku nešto manje od pet megabajta. Fajlovi se raspoređuju u nekoliko direktorijuma, a ceo paket smešta sa u jedan jedini glavni direktorijum pod nazivom MASM. Modifikacijom datoteka AUTOEXEC.BAT može se automatski prilagoditi staza za pristup svim programima.

Korisnici makroasemblera će pomalo biti iznenađeni (i uvređeni) promenom imena njihovog omiljenog programa. Umesto MASM sada

NI NALIK NA STARO

treba kucati ML, sa potpuno drugacijim (novim) parametrima. Istina, na disku postoji i program MASM, ali samo za „prvo vreme", radi kompatibilnosti sa ranijim verzijama. Uostalom, program ML nudi asembliranje, različite optimizacije i linkovanje u isto vreme.

Poseban specijalitet paketa predstavlja najnoviji Majkrosoftov „hit", program PWB (*Programmer's WorkBench*), namenjen kompletnom razvoju aplikacija, počev od editovanja izvornog koda, pa zaključno sa linkovanjem i formiranjem izvršne EXE verzije. Interesantno je da isti program ulazi i u pakel C kompajlera 6.0, čime su se asembler i C toliko približili, da bi ih u budućnosti trebalo očekivati u jednom jedinom paketu. Ova analogija dva raznorodna jezika ide mnogo dublje od PWB nivoa. Videćemo u daljem lakstu da asembler poprima mnoge karakteristike viših jezika (a naročito C-a), što posebno postaje uočljivo na polju tipova podataka, programskih ciklusa i procedura.

CELOBROJNI TIPOVI PODATAKA

Mikroprocesorska familija 8086 po prirodi stvari operiše isključivo sa *celobrojnim* podacima koji se, zavisno od dužine, mogu opisati nekim od sledecih tipova

BYTE (ili **DB**)	Neoznačen broj u opsegu od 0 do 2^8-1 (jedan bajt)
SBYTE	Označen broj u opsegu od -2^7 do $+2^7$-1 (jedan bajt)
WORD (ili **DW**)	Neoznačen broj u opsegu od 0 do 2^{16}-1 (dva bajta)
SWORD	Označen broj u opsegu od -2^{15} do $+2^{15}$-1 (dva bajta)
DWORD (ili **DD**)	Neoznačen broj u opsegu od 0 do 2^{32}-1 (četiri bajta)
SDWORD	Označen broj u opsegu od -2^{31} do $+2^{31}$-1 (četiri bajta)
FWORD (ili **DF**)	Neoznačen broj dužine 48 bita (šest bajtova)
QWORD (ili **DQ**)	Neoznačen broj dužine 64 bita (osam bajtova)
TBYTE (ili **DT**)	Neoznačen broj dužine 80 bita (deset bajtova)

Tip FWORD predviđen je za adresiranje na procesorima 80386/486, a tipovi QWORD i TBYTE imaju najčešću primenu pri radu sa koprocesorom iz familije 8087.

Asembler podržava i niz drugih, složenijih tipova podataka, ali se svi oni u krajnjoj liniji svode na pomenute celobrojne tipove.

MATRICE I NIZOVI

Matrice i *nizovi* (višedimenzione promenljive) formiraju se zapravo kao skupovi većeg broja podataka istog celobrojnog tipa. Tako, na primer, deklaracija

```
indikator  BYTE 00h
```

definiše celobrojnu promenljivu *indikator* dužine jedan bajt i dodeljuje joj početnu vrednost nula. Slično tome, deklaracija

```
niz        BYTE 01h,02h,03h
```

definiše niz od tri bajta, sa početnim vrednostima elemenata 01h, 02h i 03h. Proizvoljan *n*-ti element niza može se zadati opisom *niz*[*n*] ili *niz*+*n*. Mašinski programen vec dobro znaju da je ovakvo indeksiranje nizova samo uslovno, jer indeks *n* uvek predstavlja *broj bajtova* od početka niza, bez obzira na tip elemenata i broj dimenzija. Drugim rečima, ako su elementi niza doži od jednog bajta, ili ako smo zamislili da niz ima više od jedne dimenzije, indeks zadatog elementa moramo unapred sračunati.

Asembler nam nudi dodatni komfor u radu sa nizovima, u vidu operatora LENGTHOF, SIZEOF i TYPE. Prvi daje broj elemenala niza, drugi daje ukupan broj bajtova, a treći vraca tip podataka (broj bajtova po elamentu niza). Za proizvoljan niz *niz* uvek će važiti relacija

```
LENGTHOF niz * TYPE niz = SIZEOF niz
```

Nizovi karaktera tretiraju se prosto kao nizovi bajtova. Naravno, umesto navođenja heksadekadnih kodova pojedinih slova, može se kucati sam tekst, kao na primer

```
tekst      BYTE "Ovo je jedna poruka"
```

sto je ekvivalantno nizu od davetnaest bajtova.

STRUKTURE I UNIJE

Strukture i *unije* (tipovi STRUCT i UNION) su složeni tipovi podataka preuzeti praktično iz C-a, mada se pod drugim imenima sreću i u drugim programskim jezicima višeg nivoa. Struktura je skup podataka različitog tipa, koji po nekom kriterijumu čine logičnu celinu. Na primer, ima studenta i broj indeksa mogu se povezati u strukturu

```
STUDENT  STRUCT
         ime       BYTE  20 DUP(20h)
         indeks    WORD?
STUDENT  ENDS
```

Ovim je prakticno uveden *novi tip* podataka pod nazivom STUDENT, ukupne dužine 22 bajta, od čega prvih dvadeset predstavlja ime a preostala dva broj indeksa. Kada negde u programu treba definisati konkretnog studenta *student1*, uradićemo to deklaracijom

```
student1 STUDENT  ('Petar Petrovic 175)
```

Ako nam zatreba samo broj indeksa studenta *student1*, sve što treba uraditi svodi se na sledece

```
mov   AX,student1 indeks
```

Tip UNION je veoma sličan tipu STRUCT, s tom razlikom sto sva polja unije dele zajednički memorijski prostor. Na primer, unija

```
DWB  UNION
d    DWORD  ?
w    WORD   ?
b    BYTE   ?
DWB  ENDS
```

zauzima četiri bajta, koliko je potrebno najdužem polju *d*. Niže dva bajta istovremeno su re-

zavisna za polje *w* a najniži za *b*. Ovakva organizacija omogućava da se nekoj promenljivoj tipa DWB pristupa po želji kao 8-bitnom 16-bitnom ili 32 bitnom podatku.

Operatori LENGTHOF, SIZEOF i TYPE mogu se primenjivati i na strukture i unije da bi se dobila tražena informacije o veličini i tipu promenljive.

Interesantno je da se strukture i unije mogu definisati jedne unutar drugih. Programer će možda malo laza da izađe na kraj sa tako složenim tipovima podataka, ali makroasembleri neće imati nikakvih problema.

ZAPISI

Zapisi (tip RECORD) su takođe slični strukturama, jedino što se sada sve definiše na nivou *bitova*. Svako polje zapisa jeste ustvari niz bitova proizvoljne zadate dužine. Lep primer može da bude atribut za boju pri ispisu karaktera na ekranu. Ukupna dužina zapisa je osam bita pri čemu bit 7 daje informaciju o treptanju (*blink*), bitovi 4, 5 i 6 sadrže boju pozadine (*background*), bit 3 definiše intenzitet ispisa (*intensity*), a bitovi 0, 1 i 2 određuju boju samog teksta (*foreground*). Deklaracija ovakvog zapisa imala bi sledeći oblik:

```
COLOR   RECORD blink:1,back:3,inten-
se:1,fore:3
```

U tom slučaju atribut za normalan ispis u monohromatskom režimu (svetla slova na tamnoj pozadini) može se definisati sa:

```
normal  COLOR   {0,0,0,7}
```

Osim operatora LENGTHOF, SIZEOF i TYPE, pri radu sa zapisima mogu se koristiti još i WIDTH i MASK. Operator WIDTH daje broj bitova rezervisanih za dato polje zapisa. Na primer, WIDTH *back* će dati vrednost 3. Operator MASK formira masku za izdvajanje pojedinih polja zapisa. Recimo MASK *back* ima binarnu vrednost 01111000 a NOT MASK *back* daje invertovanu vrednost 10001111. Ako na primer želimo da izdvojimo samo boju pozadine pri normalnom ispisu, izvršićemo sledeće naredbe:

```
        mov     ah,normal
        and     ah,MASK back
```

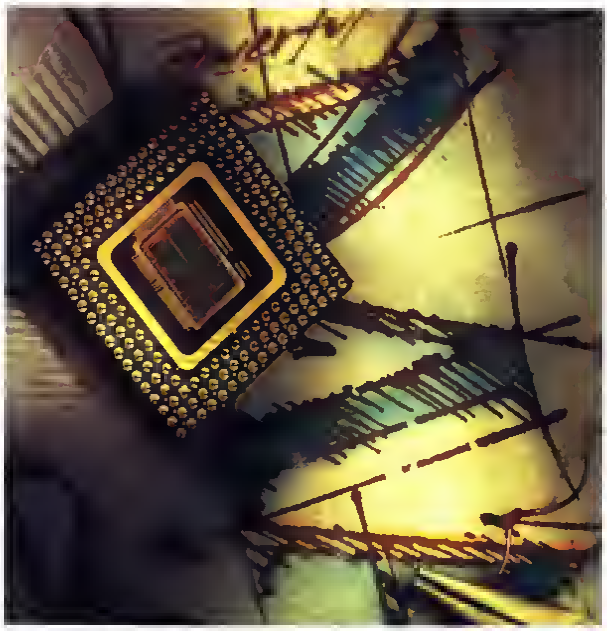
REALNI BROJEVI

Osim celobrojnih podataka, makroasembleri podržava rad i sa realnim brojevima u formi pokretnog zareza, što naročito dolazi do izražaja pri radu sa matematičkim koprocesorom. Predviđena su tri tipa realnih brojeva: REAL4, REAL8 i REAL10, što su standardne oznake za jednostruku tačnost, dvostruku tačnost i interni zapis koprocesora iz familije 8087. Ranije verzije makroasemblera su za ovu svrhu koristile deklaracije DD, DQ i DT, što je neravno izvodljivo i u verziji 6.0, ali tada nije baš jasno da li smo time definisali realan ili ceo broj. Ta nedoumica će postati očigledna kada svoj program propustimo kroz *CodeView*. Tipovi DD, DQ i DT postaće ono što i jesu – celobrojni i kao takvi će nam biti prikazani na ekranu.

Na kraju ovog prikaza tipova podataka napomenimo i to da korisnik ima mogućnost definisanja sopstvenih tipova, konkatenacijom naredbe TYPEDEF. Tako je, recimo, moguća ukucati:

```
char    TYPEDEF  BYTE
longint TYPEDEF  DWORD
float   TYPEDEF  REAL4
double  TYPEDEF  REAL8
```

čime se uvode tipovi podataka kao u C-u.



PROGRAMSKI CIKLUSI

Skromne mikroprocesorske mogućnosti za formiranje programskih ciklusa (naredba LOOP i razne vrste uslovnih skokova) makroasembleri dopunjuju WHILE i REPEAT strukturama, čime mašinsko programiranje zalazi duboko u vode viših programskih jezika.

Ideja je izuzetno jednostavna. Programer ukucava nešto poput:

```
        mov     dx,3
        WHILE   (dx != 1)
        dec     dx
        ENDW
```

gde operator != znači „nije jednako", a asembler generiše kod kao da je ukucano:

```
        mov     dx,3
        jmp     testexit
count   dec     dx
testexit cmp    dx,01h
        jnz     count
```

Programer je pošteđen izmišljanja imena za labele, a sam tekst programa je daleko pregledniji. Interesantno je koliko su tvorci makroasemblera vodili računa o optimizaciji. Ukoliko umesto uslova (dx != 1) zadamo (dx != 0), asembler neće generisati naredbu CMP DX,00H već ono što bi i dobar programer uradio, OR DX,DX.

Dok WHILE naredba proverava zadati uslov pre nego što se samo telo ciklusa jednom izvrši, REPEAT struktura vrši proveru po obavljenom ciklusu. Ako izvorni kod glasi:

```
        mov     dx,3
        REPEAT
        dec     dx
        UNTIL   (dx == 1)
```

gde operator == znači „jednako je", a asembler će preurediti kod u nešto poput:

```
        mov     dx,3
count:  dec     dx
        cmp     dx,01h
        jnz     count
```

Postoji i varijanta REPEAT ciklusa sa izlazom UNTILCXZ koja umesto uslovne *jump* naredbe na kraju generiše uslovnu *loop* naredbu, tako da se ciklus izvršava ili dok zadati uslov na bude zadovoljen, ili dok brojač CX ne dostigne nulu.

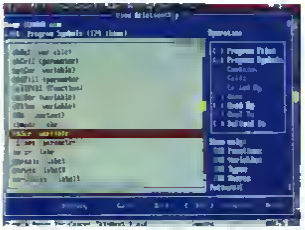
Naredba BREAK može se koristiti za izlazak iz WHILE ili REPEAT petlje po posebnom zahtevu, dok CONTINUE služi za vraćanje kontrole na početak petlje, sa bilo kog mesta unutar ciklusa.

Uslovi za testiranje obuhvataju sve operacije poređenja (jednako, različito, veće, manje, veće ili jednako, manje ili jednako, logičko NE, logičko I i logičko ILI), kao i stanja indikatora (ZERO, CARRY, OVERFLOW, SIGN i PARITY). Zaista, teško da postoji nešto što bi nam zatrebalo, a da ovim nije predviđeno.

POTPROGRAMI

Pozivanje potprograma naravno ostaje onako kako to mikroprocesor obavlja naredba CALL, smešta povratnu adresu na mašinski stek, a naredba RET skida tu adresu i nastavlja izvršanje programa. Međutim, na polju prenosa parametara ima dokta noviteta. Naredbe PROTO i INVOKE omogućuju da se definiše lista ulaznih i izlaznih paramalara u neki potprogram, i da se zatim automatski obavlja prenos parametara preko steka pri svakom pozivu.

Prenos parametara može da se izabere tako da odgovara standardu bilo kog višeg programskog jezika, što delje pojednostavljuje ve-



***Source Browser*, jedna od alatki u okviru *Programmer's WorkBench*-a, obezbeđuje obilje informacija o programu, između ostalog i kakve su veze između fukcija, promenljivih, tipova i makroa**

zu pojedinih modula između mašinskog i viših nivoa. Na primer, ako se opredelimo za C standard, parametri potprograma će biti smešteni na stek onako kako bi to uradio i C kompajler. Isto važi pri izboru paskal, fortran, bejzik, ili nekog drugog standarda. Naredbom PROTO definiše se, kao u C-u, prototip funkcije, odnosno lista ulaznih i izlaznih parametara, a pozivanje potprograma obavlja se komandom INVOKE, koja zamenjuje CALL. Uloga INVOKE naredbe je da

- Prevede argumente u željeni tip
- Prenese argumente na stek u ispravnom poretku
- Očisti stek po povratku iz potprograma.

Osim toga, sama PROC komanda omogućava da se definiše lista svih procesorskih registara koji se koriste u potprogramu, i koje bi stoga trebalo sačuvati. Asembler će u tom slučaju automatski generisati odgovarajuće PUSH naredbe na početku i POP naredbe na kraju procedure. Programi mnogo dobijaju na preglednosti, a da ništa pri tome ne gube na efikasnosti. Mnoge stvari je bolje prepustiti asembleru, nego peljati se sa njima. Asembler, recimo, sigurno neće zaboraviti da skine sa steka nešto što je sam stavio. Za programera se to ne može sa sigurnošću tvrditi.

MAKRO NAREDBE

Makro naredbe nisu specijalnost samo verzije 6.0 (ustalom i ranije verzija su nosile naziv *makroasembler*), ali treba istaći makar neke nove mogućnosti na tom polju. Recimo, uvođenjem deklaracija VARARG može se zadati promenljiv broj parametara, naredbama WHILE, FOR i REPEAT ostvaruje se ciklična struktura, a naredbom GOTO bezuslovni skok na neku označenu liniju.

Za one otrcane koji su manje upućeni reći ćemo da makro naredbe praktično služe da bi zamenile niz programskih linija, ili bilo kakav drugi tekst koji se pojavljuje na više mesta u programu. U neku ruku, makro može da zameni potprogram, s tom razlikom što asembler pri svakom pozivu umeće odgovarajući kod u tok programa.

Tipičan primer makro definicije bi bio

```
show    MACRO   char
        mov     ah,02h
        mov     dl,char
        int     21h
        ENDM
```

čime smo uvali naredbu *show* koja ispisuje zadati karakter na ekranu, koristeći standardnu DOS funkciju 02H. Na primer, možemo bilo gde u programu pisati neku od sledećih linija

```
show    cl
show    [bx]
show    'A'
show    slovo
```

U poslednjoj liniji primera podrazumeva se da smo unapred definisali promenljivu *slovo* tipa BYTE.

Naravno, definicija makro naredbe može biti i daleko složenija, ali je princip uvek isti: postoji lista ulaznih parametara koje asembler prosto prepisuje na odgovarajuća mesta i umeće zadati kod u programa. Izvorni tekst postaje pregledniji, a programer štedi trud i vreme.

Macro Assembler 6.0 pruža i nekoliko već definisanih makroa za rad sa stringovima (izvlačenje podniza, preuzrazivanja, dužina niza, nadovezivanje nizova). Sve ova operacija moguće je vršiti i na nivou komandi asemblera.

POMOĆNI PROGRAMI

Teško da danas ima razvojnog programskog paketa za neki jezik, a da u njega nije uključena neka verzija programa MAKE. To je program koji olakšava sistemske poslove oko asembliranja i linkovanja pojedinih modula, vodeći računa, između ostalog, o datumima kreiranja datoteka i sličnim sitnicama koje bi konsiniku samo oduzimale vreme.

Macro Assembler 6.0 isporučuje se sa programom NMAKE, koji se donekle razlikuje od slične verzija MAKE, o čemu se svi detalji mogu naći u detaljnom uputstvu za korišćenje.

Program NMAKE zahteva od korisnika da na početku razvojnog procesa definiše sve datoteke koje sačinjavaju projekat i način na koji su povezane. U slučaju da dođe do izmene samo neke (ili nekih) datoteka, NMAKE će automatski obaviti sva neophodna poslova da bi formirao novu EXE verziju, uzimajući u obzir nastale promene.

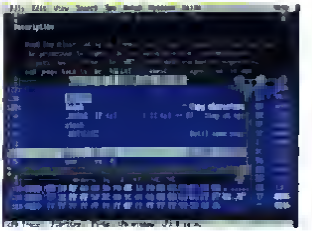
Može se to naravno, raditi i nekom BAT datotekom, ali NMAKE je daleko bolji izbor, ako ni zbog čega drugog, ono samo zato što on asemblira samo one datoteke koje su promenjene u međuvremenu. Osim toga, NMAKE nudi i niz drugih pogodnosti (kao što su makroi) koje pojednostavljuju razvoj složenijih projekata.

Još veći komfor pruža integrisani program PWB, koji omogućava kompletan razvoj i testiranje projekata bez povratka u operativni sistem. Jednostavnim izborom iz priloženog menija možemo ući u kvalitetan editor za pisanje izvornog koda, zatim ćemo tekst snimiti na disk, pozvati NMAKE, asemblirati i linkovati module, a na kraju testirati i izvršiti program.

Prednosti rada sa PWB programom je veoma obiman i kvalitetan HELP za golovo sve što radimo. Nudi nam se čak i poseban program HELPMAKE za prošinvanje HELP datoteka, ili kreiranje novih. U krajnjoj instanci možemo olici dolje da nam u toku rada ne zatreba ni jedna jedina knjiga. Bilo koja informacija o naredbama mikroprocesora, komandama asemblera, linkera ili drugih pomoćnih programa, podaci o BIOS i DOS interaptima i slično postaće dostupni pritiskom na taster.

Korisne adrese

Microsoft GmbH
Eastern Europe Department
Edisonstrasse 1, 8044 Unterschleissheim
Tel 089/31705-453, Fax 089/31705-598
Atlantis d o o
Cankarjeva 4, 61000 Ljubljana
Tel/Fax 061/ 221-606



***CodeView* dibager predstavlja izuzetno moćnu alatku za otkrivanje grešaka u programu. DOS aplikacija može biti bilo koje veličine, a debagovanje velikih programa se može vršiti i na običnom AT 286 računaru**

MOĆAN DISASEMBLER

Razvoj programa bez mogućnosti testiranja i traženja grešaka apsolutno je nemoguć. Testiranje programa primitivnim alatima je, sa druge strane, mučno. Zato uz svoj makroasembler, Majkrosoft nudi i svoj dibager *CodeView*, koji se u manje-više istoj formi može sresti u svim programskim paketima za jezike višeg nivoa.

Osnovne mogućnosti dibagera su

- Pregledanje i izmena sadržaja promenljivih i memorije
- Potpuna kontrola toka izvrsavanja programa
- Efikasna tehnike traženja grešaka

CodeView deli ekran na zasebne logicna celina nazvane prozorima. Svaki prozor (*window*) funkcioniše nezavisno od ostalih

- *Source window* prikazuje izvorni tekst programa, ili bilo koju drugu izabranu ASCII datoteku
- *Commend window* služi za ukucavanje komandi sa tastatura
- *Watch window* daje sadržaj izabranih promenljivih
- *Local window* prikazuje sadrzaja lokalnih promenljivih za tekuci potprogram
- *Memory window* daje sadržaj memorije u okolini izabrane adresa
- *Register window* prikazuje stanja svih registara mikroprocesora
- *8087 window* prikazuje stanja svih registara matematickog koprocesora ili odgovarajuceg emulatora

Moguce je u bilo kom trenutku promeniti veličinu bilo kog prozora, ili čak ukloniti nepotrebne prozora u tom trenutku.

Izvrsavanjem programa korak po korak može se pratiti svaka akcija i uočiti greška. Preko provarenih delova programa moze se prelaziti brže zadavanjem prekidnih tačaka.

VEZA SA VIŠIM JEZICIMA

Teško da se u jednom tekstu ovog formata mogu navesti sve karakteristika programskog paketa čije uputstvo broji vise od 800 strana (!). Posebno poglavlja u knjizi predstavlja i protokol povezivanja mašinskih modula sa programskim jezicima višeg nivoa, kao što su C, fortran, paskal, ili bejzik. Detaljno uputstvo sa primerima za svaki od pomenutih jezika više su nego dovoljni za brz i komforan rad. A kada se jednom sve opisane tehnike nauče, neće biti teg programerskog izazova koji ne bismo bili u stanju da savladamo.

VEŠTINA ČITANJA

Iako su odavno postali punoletni, računari se ne snalaze baš sjajno sa čitanjem – ako imate dokument na papiru sa podacima koje treba obraditi na računaru, možete se odlučiti za sporo i skupo prekucavanje ili pokušati sa nekim OCR (*Optical Character Recognition*) programom koji bi, uz pomoć skenera, trebalo da automatizuje unos podataka. Na tržištu ima više programa ovoga tipa, ali se većina „posmatrača" slaže da je među njima ubedljivo najbolja *Recognita* (odnosno, u poslednje vreme, *Recognita Plus*) mađarske firme *SzKI Recognita Corporation*. U okviru ovog testa pokušaćemo da utvrdimo da li je najjači predstavnik ove klase softvera dovoljno efikasan i za praktičnu primenu.

Dejan Ristanović

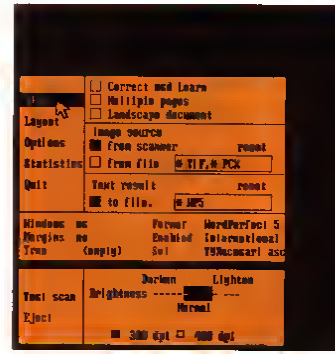
Potreba za OCR programima je nesumnjiva – živimo u svetu koji je u mnogome zasnovan na papirima i, iako ozbiljno radimo sa kompjuterima, stalno imamo potrebu da le papire nekako transformišemo u mašinski čitljivu formu kako bismo ih dalje koristili i obrađivali. Moguce je da ce ova potreba sa vremenom opasti kako kompjutera bude više – u našoj Redakciji je, na primer, golovo sav materijal koji pristiže već u mašinski čitljivoj formi. Pa ipak, ima mnogo poslova koje obavljamo sporije koji nas koštaju više ili koje, što je najgore, uopšte ne obavimo zato što je unos podataka spor, skup i nepouzdan. Dobro nam je poznato da je u drugim sredinama (posebno onima koje upravo uvode računare) problem još akutniji.



JOŠ UVEK U POVOJU

Zbog svega toga problem automatskog čitanja (OCR) već duže vreme okupira pažnju kako programera tako i korisnika računara – kada bi se omogućilo brzo i potpuno pouzdano prepoznavanje teksta, pristup računarstvu bi se gotovo iz temelja promenio. Jedino što bi moglo da izazove još značajniju „revoluciju" bila bi mogućnost pouzdanog prepoznavanja glasa. Zbog svega toga, brojni teoretičari i programeri iskušavaju svoje sposobnosti na dizajniranje OCR programa, pre svega za velike sisteme sa ogromnom procesorskom moći i memorijskim kapacitetima (npr. superkompjutere), a poslednjih godina i za svima pristupačne personalce. OCR programi za PC nisu nikakva retkost, ali je njihov kvalitet često više nego sumnjiv. Tu, pre svega, mislimo na programe koji se besplatno isporucuju uz skenere (naročito ručne) i koji bi trebalo da ubede kupca da će mu se nabavka te (ne previše skupe) sprave vrlo začas isplatiti. Kao što to obično biva, besplatan program vredi koliko i košta, pa se kupac skenera vrlo brzo oprosti od ovakvih primena.

Situacija je nešto bolja kod programa koji se prodaju nezavisno od opreme ili poklanjaju uz skupe skenere, ali preve performanse mogu da se očekuju tek od veoma skupih OCR programa; konkurencija na ovom planu je i dalje relativno mala, pa visoka cena uglavnom obećava kvalitet. Visoke cene su nas, na žalost, sprečile da neposredno isprobamo razne OCR programe; najbolji koji nam je došao do ruke bio je *ReadRight* koji je sasvim pristojan program, ali teško uporediv se *Recognitom Plus*. Pre svega, *Recognita Plus* ima nesumnjivo viši stepen „pogotka". Drugo, *Recognita Plus* je pravi poliglotski program, što svakako proizlazi iz činjenice da su ga pisali mađarski programeri, i novije verzije američkih programa u uzvesnom smislu pretenduju da budu poliglotske, ali je takva podrška brojnim jezicima uglavnom „površinska" i u mnogo čemu manjkava. Kod *Recognite Plus*, ugrađena je u same temelje programa – veoma smo se, na primer, prijatno iznenadili kada smo ustanovili da program, bez ikakvog dodatnog „učenja" prepoznaje YU slova ćčdžš.



Slika 1 Radni ekran pod DOS-om



Slika 3 Radni ekran pod *Windows 3.0*

Recognita Plus se može nabaviti u dve varijante – engleskoj i internacionalnoj. Engleska verzija je namenjena američkom i britanskom tržištu i ne podržava strana slova, dok je namenjena raspolaganju bila internacionalna varijanta koja će verovatno biti interesantna i za sve čitaoce ovog teksta. Između dve verzije je još jedna „sitna" razlika – evropska je zaštićena napravima koja se kači na bilo koji paralelni port.

Program inače stiže na dve 5.25 inča HD diskete i jednoj 3.5 inča diskeli od 1.44 megabajta, možete ga instalirati sa jednih ili sa druge, pri čemu se posao malo brže obavi u prisustvu drajva od 3.5 inča. Čitav postupak nije ni dugotrajan ni „naporan" – treba zadati direktorijum u koji se program upisuje, odgovoriti na pitanja o grafičkoj kartici, mišu i skeneru i sačekati da se datoteke raspakuju. Jednom instaliran program zauzima oko 2.5 megabajta prostora na disku, pri čemu je u ovaj zbir uračunata kako DOS tako i *Windows 3.0* verzija.

BEZ ALATA...

Kada se radi o hardverskim zahtevima, *Recognita Plus* nije baš naivna. Treba vam pre svega relativno brza mašina, u uputstvu se pominje barem AT na 16 MHz, što obično znači da je pozeljan 386 računar. Program bi zapravo verovatno mogao da se pokrene i na sporijoj mašini, ali se pojavljuju problemi pri „zahvatanju" onoga što skener literuje – sporija mašina bi zahtevala česta zaustavljanja glave skenera a to baš ne doprinosi kvalitetu čitanja. Što se memorije tiče, teorijski minimum je 640 K (ako hoćete da program radi i pod *Windows*-ima, treba vam megabajt), ali je u praksi za komforan rad potrebno 2-4 megabajta RAM-a. *Recognita Plus*, naime, obrađuje podatke „interaktivno" dok pristižu sa skenera – ako joj je na raspolaganju dovoljno memorije „pokupiće" čitavu stranu i onda je „pročitati" što u praksi radi izuzetno brzo i pouzdano. Ako memorije za tu stranu nema „čitenje" se mora obavljati „u hodu" što usporava stvar i umanjuje kvalitet. Program smo testirali na „klasičnom" 386 računaru (33 MHz, 64 K *cache*) sa 8 megabajta RAM-a čiji je višak (dve megabajta smo potrošili na *PC Kwik*) pomoću 386MAX-a konfigurisan kao XMS, ukoliko *Recognitu Plus* koristile na AT mašini u njoj možete imati bilo LIM (*expanded*) bilo *extended* memoriju.

U najvažnije komponente OCR sistema, prirodno spada i skaner – ako on loše „skuplja" tekst, *Recognita Plus* nema mnogo šanse da lakav tekst uspešno „pročita". Uz program dobijate knjižicu od tridesetak strana, zvanu *Supported Scaners* iz koje ćete naučiti da *Recognita Plus* podržava preko 60 skenera, tj. praktično sve što od ove opreme postoji na tržištu. Ili barem sve **osim** „ručnih" skenera – autori *Recognite Plus* su (rekli bismo, sa punim pravom) zaključili da ručni skaner ne garantuje kvalitet koji obezbeđuje uspešan OCR, pa su odlučili da ne slave svoju reputaciju na kocku podržavajući nešto što se objektivno ne može podržati. Pa ipak, ako imate ručni skaner i puno poverenje u svoju ruku, možete skenirati tekst, upisati rezultate u datoteku (*Recognita Plus* podržava TIF i PCX formal što u današnje vreme obezbeđuje svaki skaner) i onda tu datoteku proslediti programu – za rezultate sami odgovarate.

Što se rezolucije tiče, *Recognita Plus* zahteva 300-400 tačaka po inču, što je ujedno ono što većina današnjih skenera obezbeđuje. Pa ipak, nisu retki ni uređaji koji obezbeđuju **više** rezolucije koje, začudo, *Recognita Plus* **ne** podržava. Na prvi pogled ovo zvuči dosta nelogično – viša rezolucija svakako obezbeđuje kvalitetnija skaniranje pa bi prema tome trebala da poveća i uspešnost „čitanja"! Verovatno algoritmi ugrađeni u sam program i neka „malrice" prema kojima *Recognita Plus* prepoznaje slova „pretpostavljaju" ovakvu rezoluciju – veća bi zahtevala nove algoritme i (eventualno) brži procesor. Bilo kako bilo, na *HP ScanJet Plus* skaneru bile su nam na raspolaganju rezolucije 300 i 400 tačaka po inču, pri čemu su rezultati u svakoj od varijanti približno slični. Obzirom da su 300 dpi skaneri daleko češći, većina primera koji prate ovaj tekst dati su u toj rezoluciji.

Za sada toliko o hardveru – normalan rad *Recognite Plus* zahteva da čitav sistem bude propraćen i odgovarajućim softverom, pri čemu sa le strana ne treba ništa posebno retko: DOS 3 ili noviji (mi konstimo DOS 5.0), *Windows 3.0* (ako planirate da koristite *Windows* verziju) i drajver za skaner koji se isporučuje uz uređaj. Za docniji pregled teksta je kod DOS verzije potreban i neki tekst procesor (mi smo koristili *WordPerfect*). I na kraju – zaštita. I to dvojaka! Pre svega, senjski broj koji je toliko ogroman kao da je predviđeno da ce svaki stanovnik Galaktičkog Carstva imati po pet primeraka *Recognite Plus* broj je ugrađen u sam program i sa njim nemate nikakvog posla osim što sa u svim prilikama pojavljuje na ekranu. Drugo, hardverski modul („dongl") koji se priključuje na neki od paralelnih portova računara – mi smo ga vezali na LPT2 između računara i kabla koji vodi do matričnog štampača. Za razliku od drugih sličnih modula koji su žestoko zatopljeni, onaj koji dobijete uz *Recognitu Plus* je prilično mali i opremljen „prozorčićem" kroz koji se vidi nekakav omanji čip. Razlika u odnosu na „stare" zaštitne module sa kojima smo imali posla je i u tome što *Recognita Plus* ne zahteva da printer, ako je povezan, bude uključen, u dokumentaciji se doduše pominje da bi ovo *moglo* da bude potrebno, ali je kod nas sve savršeno funkcionisalo i kada je štampač isključen. Ljudi koji se time bave kažu da je ugrađena zaštita od „češćih" (mada se testira samo na jednom mestu...) što je i razumljivo kada se zna na kakvo tržište program može da dođe.

Pre startovanja programa morate da podesite metod koji će se koristiti pri skaniranju – ukoliko imate dovoljno memorije (npr. 4 megabajta), najbolje je da u AUTOEXEC.BAT ili u .BAT proceduru koja startuje *Recognitu Plus* ugradite SET SCANMETHOD = FAST čime obezbeđujete da skaner pročita čitavu A4 stranu i tek onda počne sa prepoznavanjem teksta. Red je na ovaj način ubeđljivo najbrži. Na „siromašnijim" sistemima treba koristiti SET SCANMETHOD = MEDIUM ili SET SCANMETHOD = NORMAL dok je SET SCANMETHOD = DIRECT rezervisano za najsporije mašine. Ukoliko promenljiva SCANMETHOD nije definisana, *Recognita Plus* na neki način sama određuje koji će metod koristiti pri čemu, bar ne našem sistemu, ovakvo prepoznavanje nije prošlo baš slavno – očigledno je biran MEDIUM metod iz razloga koji su nam nepoznati. 8 megabajta teško može brti malo za jednu stranu teksta! Tako je naš AUTOEXEC dobio još jedan SET. Ne sviđa nam se, osim toga, što se malod mora birati pre startovanje *Recognite Plus* i što se docnije na moze menjati – bilo bi veoma zgodno „počeli" veći posao u MEDIUM modu a zatim, kada sa skener malo „našlaluje" preći na FAST. Komandom SET može se podešavati i tip *high* memorije koju *Recognita Plus* koristi, direktorijum u koji sa smeštaju privremene datoteke i per drugih stvari, nama nista od toga nije bilo potrebno.

POD DOS-om I WINDOWS-ima...

Po startovanju DOS verzija programe dobijate ekran sa slike 1 kojim se veoma jedno

slavno „upravlja" – u najvećem broju slučajeva je dovoljno da stavite list papira u skaner i kliknete na READ. Teksi koji se čita sa, kako proces napreduje, pojavljuje na ekranu, da bi vam na kraju bilo ponuđeno da ga upišete u neku datoteku. Ukoliko sukcesivno skanirate veći broj listova program će ih automatski nadovezivati na kraj datoteke, ali nemate mogućnost da „iznebuha" zatražite da se tekst sa jednog lista upiše na kraj neke postojeće datoteke što je nesumnjiva slabost. Možete se, naravno odlučiti i da svaki list upisujete u posebnu datoteku, pa da te datoteke onda nadovezujete u svom tekst procesoru.

Formal izlaznog dokumenta se podešava u meniju *Options* – od važnijih formata podržani su običan ASCII, ASCII bez kolona, uravnati ASCII, generički tekst procesor, EBCDIC, DCA RFT, *DisplayWrite 5*, Dec DX, *Excel*, *Framework III*, *GSA Navy Dif*, *Samna Word*, *Lotus 123*, *Mass 11*, *Word 3.0*, *Word for Windows*, *MultiMate*, *Q&A Write*, *WordPerfect 4.1-5.1*, *WordStar 3.3-6.0* i *XyWrite* (zanimljivo da *Ventura* formal nije podržan). U početku smo koristili generički tekst procesor ali se pokazalo da na taj način „gubimo" podatke o naglašenom (**bold** i *italic*) tekstu pa smo prešli na *WordPerfect 5.1*.

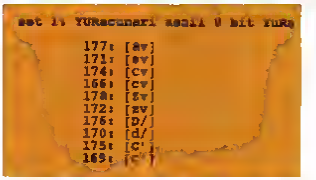
U okviru *Options* menija podešava se još jedna važna stvar, sai znakova. U startu se koristi PC „azbuka" ili eventualno *Roman 8* ali su nama bila potrebna YU slova. Na svu sreću, uz *Recognitu Plus* se isporučuje konfiguraciona datoteka prilagođena našem podnevlju koja obezbeđuje „čitanje" teksta u kome se javljaju slova čćđšž koja se prikazuju na ekranu i upisuju na disk u skladu sa YUSCII rasporedom. YUSCII baš i nije naše omiljeno rešenje ali smo, čitajući dokumentaciju, saznali da ga je vrlo lako promeniti – kreirali smo datoteku YURA.SET sa slike 2 i naša *Recognita Plus* je začas „progovorila" prema rasporedu „Računari". U istom meniju podešava se i jezik na kom *Recognita Plus* komunicira sa korisnikom (pored ostalih na raspolaganju je i hrvatski) i zadaju neki specijalni znakovi koje treba uzimati u obzir na primer grčka slova. Na prvi pogled izgleda prirodno da sa skup mogućih znakova što više proširi pošto se tako obezbeđuje prepoznavanje raznovrsnijeg teksta, ali više raspoloživih znakova povećava verovatnoću greške pa je najbolje uključiti samo ono što se zaista javija u tekstu koji se analizira. Za ekstremne prilike *Recognita Plus* se može programirati da „prepoznaje" samo cifre i par znakova (tačka, zapeta i slični), čime se smanjuje verovatnoća greške u numeričkim tabelama.

Meni *Leyout* je zadužen za definisanje dela strana koji se skenira – tako *Recognita Plus* sama razdvaja deo na kome je tekst od slika ili belina, ponekad je pogodno definisati prozore koji pokrivaju teksi koji se skenira ili (po izboru) koji se **ne** skenira. Slično tome, ukoliko skeniramo veći broj A5 strana, ima smisla delimično isecati donju i desnu marginu i tako na faraš skaner da svaki put prelazi preko nepopunjenog segmenta zamišljanog A4 lista. *Leyout* meni omogućava zadavanje većeg broja prozora i graničnika ali je ovakvo nešto u najvećem broju slučajeva nepotrebno. Pa ipak, dobro je da postoji pošto u nekim specifičnim situacijama može da spase mnogo vremena.

Meni *Statistics* najzad, prikazuje osnovna podatke o tome koliko je prethodno prepoznavanje bilo uspešno – koliko je vremena potrošeno na skaniranje a koliko na prepoznavanje, koliko je znakova prepoznato, koliko nije, koliko ime slobodne memorije i tome slično.

Ostale stavke osnovnog ekrana omogućavaju izbor orijentacije dokumenta (*Recognita Plus* može da „čita" i *landscape* materijale), definisanje izvora (skaner ili TIF odnosno PCX datoteka) i odredišta teksta, probno skaniranje i, što je najvažnije, podešavanja osvetljenosti. Na *Hewlett-Packard*-ovim skanerima može da se izabere opcija *Darken*, *Normal* ili *Lighten* – navodno je *Hewlett Packard* obezbedio čak sedam različitih intenziteta osvetljenosti ali se u svim komercijalnim programima sa kojima smo imali kontakt konste samo ova tri. Ispravno podešavanje osvetljenosti je jako važno i kod skaniranja, a kod prepoznavanja znakova postaje presudno. Iskustvo se može svesti na jednostavnu tvrdnju: ako je original dobar gotovo da je svejedno kojim ga osvetljenjem skenirate. Ako je bled, zabrljan ili (naročito) ako je tekst vili pozadina u nekoj boji, jednim osvetljenjem nećete dobiti baš ništa, drugim ćete dobiti crnu mrlju a trećim solidan i prepoznatljiv tekst. Odrađivanje ispravne osvetljenosti je veština koja se stiče kroz višegodišnju „skanersku" praksu – početnika može da ureši to što metodom probe i greške, u najgorem slučaju neku stranu skenira tri puta što i nije tako strašno. Osim osvetljenosti, neki skaneri omogućavaju i podešavanje kontrasta, ali kako naš *ScanJet* to ne podržava, nismo bili u prilici da probamo.

Windows 3.0 verzija (slika 3) je, kao što je i red, nešto bogatija od DOS ekvivalenta. Najbitnija prednost je u tome što je saslavni deo paketa i solidan editor, a naravno da sve to izgleda lepše i koristi se jednostavnije. Najveća usluga je slična – u *File* meniju možemo da učitamo neku već skeniranu TIF datoteku, iz *Recognition* menija aktiviramo čitanje i prepoznavanja teksta, u *Scan* meniju podešavamo skaner i oblast lista koja se skenira, u *Options* meniju podešavamo razne parametre dok *Window* i *Help* meniji sadrže funkcije uobičajene za sve *Windows* aplikacije. Kao što rekosmo jedina bitnija razlika je u tome što se jednom pročitani i „prepoznati" tekst, pre upisa u datoteku, prenosi u editor teksta u kome ga možemo pregledati pratražiti (meni *Search*) i ispravljati (meni *Edit*). Lepo je što se u tom editoru bez ikakvih posebnih intervencija, na du sva strana slova, pa i YU znaci. Pa ipak, editor retko ima smisla koristiti jer grešaka obično ima previše za neko „pristojno" ispravljanje – treba sve strane sklopiti u dokument a onda, u svom omiljenom tekst procesoru pažljivo „raditi" na njegovom dovođenju u željeno stanje.



Slika 2 Datoteka za YURA SET za promenu kodnog reaporeda u standard „Računari"

LIČNA KARTA

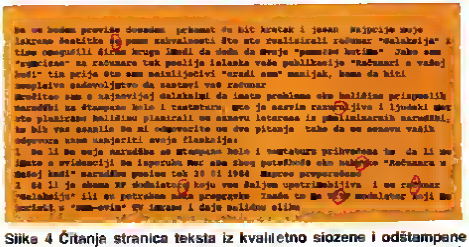
Program Recognita Plus
Verzija 1.2 25. jun 1991
Namena OCR: automatsko prepoznavanje karaktera
Sadržaj paketa Dve diskete 1.2 M jedna disketa 1.44 M tri priručnika i modul za hardversku zaštitu
Okruženje IBM AT ili kompatibilan, 1 M RAM-a (poželjno 2 M) hard disk, miš, 300-400 dpi skaner, DOS 3/Windows 3 ili noviji
Prostor na disku oko 2.4 megabajta
Ulaz Tastatura, miš, skaner
Izlaz – ekran EGA/VGA/Hercules ili bilo koja kartica koju Windows 3.0 podržava
Dokumentacija User's Guide – MS Windows Environment (160 A5)
User's Guide – MS Windows Environment (160 A5)
Supported Scaners (32 A5)
Proizvođač SzKI Recognita Corporation
Donati u. 35-45
H-1015 Budapest
Hungary
Tel.: (36-1)201-8452
Fax: (36-1)201-7606
Zastupnik LTS
11080 Beograd, Dobrovoljačka 9
Fax/Tel 011/190-572
Cena 2400 DEM ili 45000 D (krajem avgusta)

... I KOLIKO POUZDANO

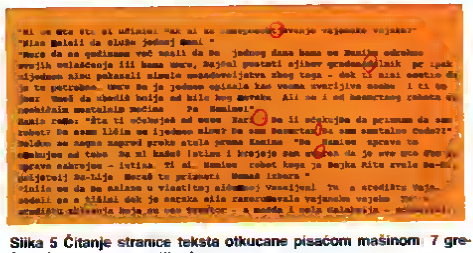
Verujemo da je sva dosadašnja diskusija pokazala da je *Recognita Plus* konsi jednostavno i afikasno čak i bez posebne konsultacija inače veoma detaljnog i lepo napisanog uputstva u dva toma (jedan za DOS drugi za *Windows*). No pravo pitanje je koliko je program zapravo afikasan i koliko grešaka pravi. Kao i kod većine drugih programa ovoga tipa, efikasnost veoma zavisi od onoga šlo „podmetnala" kao test primer.

U specifikacijama programa pročitaćete da je tačnost 99.9% čemu se posle samo par proba, možete jedino slatko nasmejati. Pa ipak malo računanja nije na odmet – broj 99.9% možete da posmatrate subjektivno i objektivno. Sa subjektivne strane reći 99.9% znači da program radi „bez greške", to je jedini smisao koji u realnom životu pripisujemo ovoj brojki! Objektivno, ako jedna štampana strana sadrži nekih 4000 znakova, jedan promil označava neka četiri greške što, ako je original kvalitetan, i nije toliko daleko od istine. Dodatna povoljna okolnost je što *Recognita Plus* ne pravi „greške u kucanju" poput ljudskih koje je lešo primetiti – ako recimo malo slovo a buda prepoznato kao broj *5* (što se zna desiti) greška će biti vidljiva „iz aviona". I mnoge druge greške su toliko „nehumane" da se relativno lako uočavaju. Pa ipak u praktičnoj primeni *Recognite Plus* u YU uslovima posloji jedna velika prepreka – ako Amerikanac uzme „pročitani" tekst i propusti ga kroz program za kontrolu pravopisa, koji se isporučuje uz sveki iole pristojan tekst procesor, pronaći će 99.9% grešaka koja je *Recognita Plus* „skrivila". Mi na žalost, program za kontrolu spelovanja nemamo pa ostaje samo da se bulji u monitor – nije to naravno „kvrica" *Recognite Plus*, ali i o toj činjenici treba voditi računa.

Da bismo ilustrovali kako *Recognita Plus* funkcioniše „podmetnuli" smo joj nekoliko vrsta teksta. Da vidimo pre svega kako je „prošlo" čitanje stranice teksta štampane na laserskom štampaču *HP LaserJet III* uz korišćenje standardnih *Bitstream* fontove (*Helvetica 10 pt*) sa YU slovima. Na strani je bilo nešto polucmog i kurzivnog teksta: zapravo svaka pojava reči *Recognita Plus* bila je ispisana kurzivom. Rezultati su umereni – program je napravio 16 grešaka i u teksi ubacio pel „kockica" kojima je zamenio znakova koje nija uspeo da prepozna; u ova greška nismo računali to što je kurziv prepoznat na samo jednom od desetak meste. Ovo nema veze sa onih 99.9% (ma



Slika 4 Čitanje stranica teksta iz kvalitetno siozene i odštampane knjige (foto-slog + tabačni ofset): 15 grešaka i 11 neprepoznatih slova



Slika 5 Čitanje stranice teksta otkucane pisaćom mašinom: 7 grešaka i 38 neprepoznatih slova

kako tu brojku shvatiti, ali u „odbranu" *Recognita Plus* treba reći da je najveći broj grešaka napravljen kod YU slova i znakova navoda (lu mačeni kao apostrofi). Što je, zapravo, dokaz da su sva slova jednaka, ali da su neka „jednakija" od ostalih – autori *Recognita Plus* jesu podrzali YU (i razna druga strana) slova, ali su ipak daleko veću pažnju posvetili ASCII znakovima!

Na slici 4 vidimo kako se *Recognita Plus* „snašla" sa knjigom štampanom u slanim dobrim vremenima, pre DTP-a. Tada su knjige, ako se sećate, štampane na kvalitetnom papiru i na foto slogu i ne na ovoj današnjoj 300× 300 rezoluciji i papiru koji obično opisujemo jednom nepristojnom reči. Rezultati ponovo nisu baš ajajni u lekstu se našlo čak 11 „kockica" koje označavaju nepreposnati znak dok je slovnih grešaka bilo 15. Kurzivni odlomci su nešto pouzdanije prepoznavani (mada daleko od savršenog; poslednji pasus je u originalu **čitav** napisan kurzivom). Primetimo, međutim, da je veliki broj slovnih grešaki i neprepoznatih slova ponovo usko vezan sa YU znacima.

Sa „modernijom" DTP knjigom na na baš slavnom papiru rezultati su bili prilično dobri, zapravo izgleda da je ovaj font više odgovarao *Recognitr Plus* od prethodnog. U lekstu amo pronašli samo šest slovnih grešaka, dok je neprepoznatih slova bilo nešto više, čak 18. Ovo ga pula problem se još jesnije polarizovao – bezmalo **sve** greške i neprepoznati znaci su YU slova.

Interesovalo nas je i kako *Recognita Plus* čita novine – pošto se sa ćirilicom još ne snalazi (ćirilične verzija je u pripremi), skanirali smo jednu vesi iz „Borbe" – 21 neprepoznat karakter i 10 slovnih grešaka u relalivno kralkom tekstu ne zvuče baš sjajno, ali ipak treba uzeti u obzir na kakvom se papiru „Borba" štampa. Probe sa „Računanma" na kunzdruk hartiji daju znalno bolje rezultate dok je sa našim starim umecima situacija jako loša. Zaprevo ako je kontrasl između leksta i pozadine dobar, prepoznavanje je dosta uspešno a ako je papir „žućkast" leksi se pre sniže nego čita. U lakvim slucajevima najznačajnije je pravilno podešavanje osvelljenosli.

Ne slici 5 smo pokušali sa pisecom mašinom – obzirom da isiu već godinama ne konslimo nije bilo lako pronaći neki primerak koji je štampan se dovoljno dobrom trakom i na relalivno novoj mešini čija slova nisu iskrzana. Najzad smo ga pronašli među nekom (pra)stalrom poštom koju amo ko zna zašto sačuvali. Kao što se vidi, rezultati su umereni – slovnih grešaka ima relativno malo (7), ali je zalo neprepoznatih znakova čak 38. Povoljna okolnost je što je nepropoznati znak uglavnom slovo j koje iz nekog razloga nije bilo „po volji" *Recognite Plus*.

Pokušaj citanja teksta sa malničnog štampača proizvodi kalastrofalne rezultate – nismo ga ni ilustrovali slikom jer bi na njoj bilo više „kockica" nego slova. To se ne odnosi samo na naš stan developinski *Canon PW1080A* (iako smo mu za ovu priliku svečano stavili novu traku) nego i na 24-pinaki *Epson* LQ 550 i LQ 800 kao i na dve *Panasonic* modele čije smo otiske dobili na uvid. Prepoznavanje je nešto bolje u reznim (*Near*) *Leftar Quality* modovima ali je i tada daleko ispod granice upotrebljivog. Očigledno je da *Recognita Plus* pn „čitanju" poviači neke konlure oko slova a malnični štampač ne proizvodi dovoljno „čvrsle" obnse za takvu analizu. Ako bi već trebalo birati jedan tip ispisa koji objektivno ne može da se pročita, verovatno bismo se najpre opredelili za malnički štampač u nekoj nadi da ono što je stampano na malničnom štampaču vec postoji u nekom računaru, pa bi se moglo dobiti u mašinski čitljivom obliku; dakle, najmanje zlo. Pa ipak, bilo bi mnogo bolje da se *Recognita Plus* snalazi i sa ovim ispisom.

Što se brzine tice, sve se uglavnom kreće u granicama pomenutim u okviru specifikacija *Recognite Plus* – izmedu 150 i 250 znakova u sekundi na 386/33 MHz računaru sa dovoljno memorije. Ukoliko u čitanje uračunale i vreme potrebno za skaniranje brzina „pada" na 80-120 znakova u sekundi. Brzina veoma zavisi od kvaliteta originala i kvaliteta pozadine – ako su slova slabije čitljiva *Recognita Plus* očilo mnogo duže „pokušava". Posebno je zlo ako je staklo vašeg skanera prljavo, pa se u delu koji **nije** poknvan lekslom pojavljuju neke mrlje koje bi mogle da liče na slova; brzina je tada drastično manja.

Vredi posebno istaći trud koji su autori *Recognite Plus* uložili da bi izlazni dokumenti bio direktnije upotrebljiv. Većina OCR programa će, naime, redove upisati u daloteku onakve kakvi su i ostaviti korisniku da ih spaja u pasu se sto ume da bude veoma dugotrajan posao. *Recognita Plus* će se se druga strana „potruditi" da prepozna pasuse i da <CR> <LF> slavi samo na kraj pasusa, a ne i na kraj svakog njegovog reda. Izvestan problem stvaraju jedino crlice na kraju rada koja označavaju reč podeljenu na slogove. Osim toga, kolone se razdvajaju tako da se u lekst upisuje najpre leva pa iza nje sledeca i tako dalje. Sve do desne; ovaj efekat se može isključiti ali je u najvećem broju slučajeva izuzetno korišćan. Ukoli ko izaberemo izlazni formal kompatibilan se nekim leksl procesorom (npr. *WordPerfect 5.1*) *Recognita Plus* će čak generisati i efekte uvlačenje prvog reda pasusa, zatim pasusa kod kojih je prvi red izbačen e ostali uvučeni (Kodovi **[indent]** [**MerRel**]) promena margina u okviru iste strane i svašta drugo. Predviđena je čak mogućnosl da se skanira i „pročita" neka tabela, e onda rezultat zapiše na disk u *Lotus* ili *Excel* formatu, spreman za dalju obradu u nekom od ovih programa. Zaista izuzelna ideja.

Sve u svemu, *Recognita Plus* je u praksi sasvim primenjiva – i dalje se može samo sanjati o tekstu koji bismo skanirali i odmah konstili ne brinući o greškama u njemu, ali je trud potreban za skaniranje, optičko prepoznavanje i docnije ispravljanje u većini slucajeva daleko manji od truda potrebnog za unos najzad, da je tasl i kucan trebalo bi takođe vršiti korekturu i unositi ispravke! Zavisno od primene u nekim slučajevima leksl koji proizvede *Recognita Plus* može biti čak i direktno upotrebljiv, ako se radilo o vrlo kvalitetnom originalu i ako nije baš neophodno da se svaka greščica ispravi.

KAKO RECOGNITA UČI

Jedna od važnih karakteristika svakog boljeg OCR programa je mogućnost „učenja" – ukoliko naložite „čitanje" nekog fonta koji je programu potpuno ili delimično nepoznat, treba omogućiti da posle izvesnog broja „početnič kih grešaka" čitanje postane sve efikasnije i efikasnije. *Recognite Plus* lo radi na najbolji mogući način – ukoliko aktiviramo *Learn* mod, svako „nejasno" slovo se ispisuje na ekranu u krupnom planu zajedno sa „kontekstom" iz koga je izvađeno (vrlo zgodan dodatak jer slovo ume da bude takvo da ga ni čovek ne može baš sigurno identifikovati) i onim što *Recognita Plus* predlaže (ukoliko program baš ništa ne zaključi o slovu, nema ni predloga). Od konsnika se očekuja da taj predlog prihvati pritiskom na Enter, otkuca korektno slovo (ili čak dvoznak), ili proglasi ispisani oblik za „dubre" tj. mrlju na papiru ili grešku pn skaniranju. *Recognita Plus* „pamti" ove odluke i u budućim sličnim situacijama primenjuje stečeno „znanje". Primenom *Learn* moda smo, na primer, vrlo lako i brzo postigli da *Recognita Plus* prepoznaje slovo j u leksu sa slike 3, a zatim smo ovo „znanje" snimili na disk (obzirom da veliki broj „naučenih" slova svakako usporava rad. *Recognita Plus* omogucava snimanje veceg broja „lekcija" u razne datoleke e zatim učilavanje samo onih koje su zaista potrebne za leksi na kome se radi) što bi nam bez sumnje pomoglo da smo imali još leksta sa iste pisace mašina. No i ovo „učenje" ima svoje granice – pokušaj da „naučimo" program *University Roman* fontu (podseća na pisana slova) dao je veoma slabe rezultate.

Šta reći na kraju? Jedan od OCR programa se u časopisu BYTE reklamirao fotografijom simpatičnih (pomalo kosih) ženskih očiju ispod koje je pisalo „jedini OCR sistem bolji od našeg". Ovakva reklama bi pre pristajala *Recognite Plus* po svemu što smo čuli i isprobali: redi se o najboljem OCR programu za personalna računare ali je njegova moć i primenljivosl i dalje veoma veoma daleka od svakog „ljudskog OCR-a", pa makar taj OCR išao u drugi razred osnovne škole. Krecemo se svakako napred ali ne baš prevelikim koracima!

TYPE 1 ILI TRUE TYPE

Proizvođači, jednako kao i korisnici, odavno su uvidjaju prednosti konturnih nad bit mapiranim fontovima. Paradoksalno je, međutim, da je najveća slabost konturnih fontova upravo potreba za njihovom rasterizacijom, odnosno pretvaranjem njihovih kontura u odgovarajući segmentni zapis – bit mapu! Ovu je konverziju potrebno izvršiti u trenutku, ili neposredno prije, samog ispisa, bilo da se radi o ispisu na ekran, pisač ili neku drugu, digitalnu, tiskarsku jedinicu.

Robert Slavetski

PostScript, grafički jezik kojega je Adobe predstavio 1984. godine, već je tada posjedovao skup jezičnih rutina za definiciju slovnih oblika koje su omogućavale njihovu reprodukciju na ispisnim sistemima različite razlučivosti. Ova je mogućnost, ujedno, i primarni razlog *PostScript*-ove ekspanzija u laserskim pisačima, te njegove kasnije eksplozivne migracije na fotoosvjetljivačke jedinice. Pa ipak, cijelo to vrijeme, Adobe je specifikacije najvitalnijih dijelova *PostScript*-a skrivao od očiju javnosti. Način rada *Type 1* formata, koji veoma uspješno vrši rasterizaciju konturnih fontove na *Post Script* printerima bio je, do nedavno, strogo čuvana tajna.

ADOBE OBJAVLJUJE

Prije nešto više od godinu dana, Adobe je promijenio svoju politiku i objavio detaljne tehničke specifikacije za *Type 1* format. Međutim, ovaj iznenadni zaokret nije slučajan. Na ovaj se, značajan, potez Adobe odlučio prvenstveno iz dva razloga. Prvo neke digitalne slovolivnice (kao npr. Bitstream) uspješno su dešifrirale *Type 1* format i neovisno o Adobeu proizvodile ove fontova. Drugo, Microsoft i Apple, vidjevši svoju šansu na ovom strogo zatvorenom tržištu, promovirali su u potpunosti otvoren font-format nazvan *TrueType*.

Adobe je ubrzo shvatio da neće spriječiti svoje korisnike da se prebace na novi, ludi model, sve dok je njegov vlastiti strogo zatvoren. Lukavo birajući koje će dijelove svoga sistema objelodaniti, Adobe je, se jedne strane, i dalje ostao relativno sakriven, a sa druge uspio je zadržati korisnika koje bi inače sigurno izgubio. Uz to, ovim kao i drugim postupcima koje provodi u okviru svoje nova politike, Adobe je znatno usporio širenje Apple-Microsoftove koncepcije.

PROBLEMI RASTERIZACIJE

Sada, kada su sve karte položene na stol, rat između ova dva font formata doživljava svoj vrhunac, a bitke se vode na tehnološkom jednako kao i na marketinškom planu. Obzirom da sam o problemima markeñinške prirode pisao u svojem prethodnom članku (Računar 74, članak „Real fontova"), sada ću se više posvetiti tehnologiji ovih fontova. Međutim, da bismo u potpunosti shvatili tehničke kontroverze, potrebno je poznavati osnove rastenzacija – procesa u kojem se vrši aproksimacija originalne, konturne forme karaktera u konačni broj fiksiranih i strogo raspoređenih točaka na ekranu monitora ili ispisanoj stranici.

Kao što vjerojatno i znadete, najveća prednost konturnih fontova nad bitmapiranima sastoji se u tome da oni nisu specifični za svaku veličinu zasebno, već ih je moguće skalirati drugim riječima, u mogućnosti smo reproducirati slova različitih veličina na osnovu samo jednog konturnog mastera. Međutim, kolikogod ova promjena veličine činila jednostavnom i logičnom, ona je puna zamki i uslinu je nepredvidiva. Suprotno onome što bi čovjek očekivao, nemoguće je kreirati manju ili veću verziju slova jednostavnim skaliranjem osnovnog mastera. Odnos širina/visina ne mijenja se uvijek proporcionalno sa povećanjem ili smanjenjem slovnog oblika. Suprotno tome, on može znatno varirati. Uz to, određeni dijelovi karaktera – posebice, trupovi i senfi – zahtijevaju poseban tretman. Ova nelogične distorzije čudno ali istinito, oplemenjuju krajnji izgled karaktera. Iskustva su pokazala da ljudskom vizualnom sustavu ljepše i sređenije djeluju karakter kod kojih je faktor skaliranja dobiven neravnomjernim odnosom vertikalne i horizontalne veličine, nego oni kod kojih je ovaj postupak proveden sistemom proporcionriranja.

Očito je, dakle, da ljudsko opažanje može biti izuzetno varljivo iz tog razloga, jedan od najsuptilnijih aspekata kreatorova rada vezan je upravo za kompenzaciju ovih problema. Kao što je rekao John Colins, osnivač Bitstreama: „Sam bit tipografije leži u preciznosti i regularnosti". Pa ipak, paradoksalno je da slova moraju biti izrađena neregularno kako bi ih ljudsko oko prihvatilo kao regularne. Na primjer, u lontu konštenom pri lisku ovog članka, donji dijelovi slova „o" i „t" u riječi „font" neznatno prelaze ispod slova „n" (Ako ne vjerujete, uzmite povećalo i ravnalo i uvjerite se sami.) Da slova u ovoj riječi nisu razmještena na ovaj način, dobio bi se osjećaj da slova „o" i „t" lete. Kao rezultat, riječ „font" izgubila bi na svojoj kompaktnosti i čvrstoći.

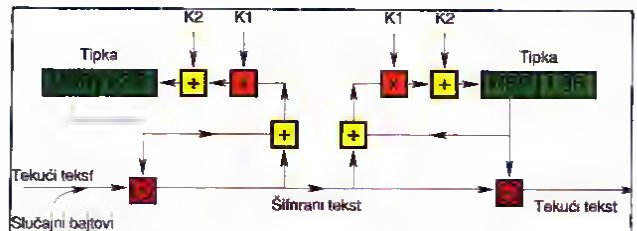
Slovni oblici mogu se predstaviti pomoću rastera (bit-mapirani fontovi), vektora (vektorski fontovi), ili kontura (konturni fontovi).

Osnovni postupak pri kreiranju odgovarajućeg karaktera iz konturnog fonta je slijedeći: skaliranje konturnog mastera do traženo veličine i prikaz onih točaka čiji centri padaju unutar kontura. Obzirom da ovaj jednostavan pristup ne računa na probleme ljudskog opažanja, on može izazvati ružne i neočekivane rezultate.

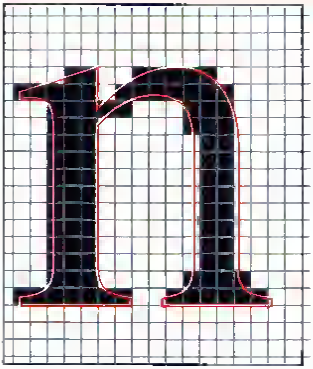
Slika 2 prikazuje osnovni problem skaliranog konturnog fonta. Kao što možete uočiti, trupovi (vertikalne osnovice) malog slova „n" rasterizirarog u izrazito niskoj rezoluciji, različiti su po debljina, prouzrokujući na taj način ružnu asimetriju. Konačno, veoma je važno da similarni karakteri istog fonta budu rasterizirani na isti način. Ukoliko „m" nije istog oblika kao i „n" u istom fontu, čitalac će odmah uočiti grešku.

Prema tome, želimo li konturne fontove koristiti za kvalitetan ekranski ili printerski ispis moramo od računala napraviti esteiski samosvjesnu jedinicu. Da bi ono napravilo simetrične trupove jednakih širina, serife istih dužina i ostale značajke koje jedan font čine upravo tim fontom, moramo smisliti način na koji ćemo računalu moći reći više o traženom, finalnom rezultatu. Ovaj je postupak poznat pod imenom predvidenja (hinting) ili pegljanje i upravo je on srce moderne font tehnologije.

Na tržištu je danas prisutan velik broj sistema za rasterizaciju i predvidenje. Među najpoznatijima su *Intellifont* (AGFA CG), *Fontware* i *Speedo* (Bitstream), *FontMaker* (Sun), *Nimbus* (URW), *Nimbus Q* (The Company), Adobe *Type 1* (Adobe), i *TrueType* (Appla i Microsoft). U daljem ću se teksi koncentrirati na *Type 1* i *TrueType*.



1. Adobe *Type 1* kriptografski algoritam koristi pseudo-slučajni brojčani generator čije izlazno stanje ovisi ne samo o vrijednosti neposredno pritisnute tipke, već i o sadržaju tekućeg teksta. „Slučajno" generirani bajtovi pridruzuju se običnom tekstu prije samog kriptografskog postupka. Na ovaj način, rezultirajući, šifrirani tekst bit će uvijek različit. U postupku dekodiranja, dakoder mora jedino znati koliko je ovih, slučajnih bajtova, koje mora ukloniti.



2 Korištenjem jednostavnog (neivnog) sistema za rasterizaciju konturnih fontova dolazi se do neočekivanih i nadasve ružnih rezultata. Kod ovog malog slove „n" skalirana kontura se nije pokiopila se rasterom, što je uzrokovalo pojavu trupova i serifa različitih širina

TYPE 1 FONTOVI

Adobe *Type 1* fontovi (ili tont programi, kako ih Adobe naziva) konste jednostaven akup *PostScript* instrukcija za crtanje kontura pojedinog slova, te dvorazinski sistem pegianja. Dok su instrukcije kojima se crtaju konture ustinu dio standardnog *PostScript* koda, sistem pegianja je, po svojoj pnrodi, prvenstvano deklarativan – točnije, on daje informacije kodu za rastenzaciju, all sam nije dio tog koda. Program koji vrši funkciju rasterizacije, a koji može biti dio *PostScript* interpretera ili zasebna programska jednica (kao u slučaju Adobe Type Menagera), naziva se *Type 1 BuildChar*.

Type 1 fontovi koriste nekoliko informacijskih nivoa koje *BuildChar* može konstiti pri izgradnji tonte. Najviši nivo daje globalne informacija vezane uz tont. To su, dakle, podaci koji se odnose na traženu visinu malih i velikih slova u tontu, kao i idealnu šinnu trupova (posebno važnu za mala slova). Ovaj nivo također sadrži i informacije koja pomažu rastenzatoru da fontovi istih porodica (npr. Times i Times Bold) nalikuju jedni drugima.

Sljedeći nivo sadrži kodove koji definiraju sam karakter. Ove male rutine, nazvane *Charstnngs*, sadrže 24 *PostScript* naredbe što je teman dovoljno za iscrtavanje slovnih kontura. Dok jedne direktno crteju linije i krivulje, druge samo pozivaju podrutine koje te funkcije izvršavaju umjesto njih. Većina ovih naradbi brojeve na steku koristi kao argumente i sve osim jedne, pomake definiraju ne kao apsolutne već kao relativne. Na ovaj je način instrukcija lakše povezati u rutine koje je zatim moguće konstiti u tontovima.

PEGLANJE

Najniži nivo procesa kreacije karaktera u *Type 1* tont formatu je peglanje (ili kao što sam već spomenuo – predviđanje). Postupak peglanja ovdje je baziran na konceptu trupova. Svaki karakter *Type 1* fonta sastoji se od vertikalnih linija (zvanih vertikalni trupovi) i horizontalnih linija (horizontalni trupovi). Dvije najbitnije linije slova „T" smatraju se trupovima toga slova, ali trupovima se smatraju i vršni krajnji serifi. Interesantno: trupovi u *Type 1* fontu ne moraju biti okomiti, vodoravni, pa čak niti pravocrtni. Donji i gornji dijelovi slova „O" uzimaju se kao horizontalni trupovi, dok krajnje lijevi te krajnje desni dijelovi predstavljaju vertikalne trupove.

Peglanje, odnosno koncepcije bazirena na trupovima omogućava sistemu rastenzacije da na lak način usienovi idealnu širini slovnih linija, te da na taj način kompenzira moguće greška u debljini. Ovaj sistem također daje podatke o gornjem i donjem dijelu karaktera, olakšavajući na taj način smještaj karaktera na odgovarajuću lokaciju.

Slika 3 prikazuje kako se sistem trupova može primjeniti na serifnom slovu „T". Važno je uočiti da su trupne zone predočene parom x- i y- vrijednosti i da se protežu duž cijelog karaktera. Ako slovo ne posjeduje serife, dizajner svejedno mora dodati popratna, nepostojeće trupove, koji će kasnije pomoći rasterizatoru da pronađe donji i gornji dio karaktera (pogledajte sliku 4).

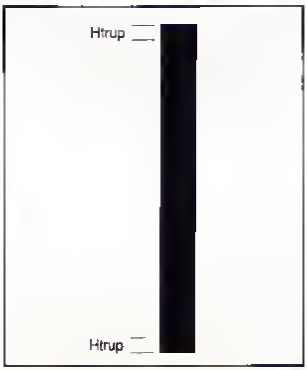
Adobeov *Type 1* font tormet posjeduje još dva dodatna sistema za peglanje: prvo, (*dotsection*) peglanje omogućava finalno rendaranje istočkanog karaktera, dok drugo (*flex*) peglanje, ravna deformirane zavijutke i krivulje, koje se često pojavljuju na niskim razoluciia ma, stvarajući zupčasta ružne slove.

Međutim postoji i sasvim druge koncepcija koju je moguće koristti za dotjenvanje skaliranih i rasterizlranih fontova: hibridni fontovi. Za razliku od običnih, konturnih fontova kod kojih postoji samo jedan master, hibridni tontovi posjeduju dva kontuma mastera. Sistem za rastenzaciju odebire jednog od dva mastera ovisno o rezolucij izlazne jedinice. Naima kontur ni masten kreirani u visokoj rezoluciji (*PostScript* definira konture fontova u veličini od 1000 točaka) sadrže sitne detalje koje na niskim razolucijama nije moguća prikazati. Primjer hibndnog tonta je Adobeov Optima font.

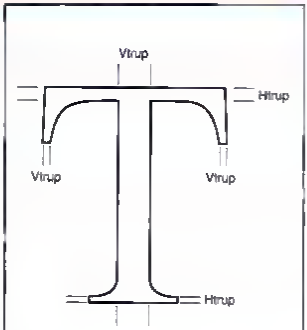
Kolika je, dakle, ovaj Adobeov sistem dobar? Očito da je dovoljno dobar za tisuće zadovoljnih korisnika *PostScript* printera. Ali ipak, tipo-dizajneri se žale da ne-latinski i ideografski karakteri (npr. Symbol) dakle oni koji nema ju klasične i jednolične trupove, nisu u najboljoj mjeri posluženi od strane Adobeovog *Type 1* mehanizma. Adobeova je direktna preporuka da se izrazitno kompleksni simboli jednako kao i logotipovi kreiraju u Type 3 tont formatu. Ovaj, naime svako slovo definira kao i svaki drugi konturni oblik i za generiranje karaktera koristi standardne *PostScript* instrukcije. Na žalost, Type 3 fontovi ne rade sa Adobe Type Managerom te ih je iz tog razloga nemoguće koristti kao ekransko/prinierske fontove na sistemima sa ugrađenim ATM-om. Uz to rasterizacija ovih fontova je osjetno duža nego što je to kod *Type 1* formata. Neposljetku, *Type* 3 fontovi nemaju ugrađen mehanizam za peglanje. Jugoslavenski dizajneri koji kreiraju ćirilične fontove, imaju sreću da ćirilica izuzetno nalikuje latinici, pa iz tog razloga nemaju nikakvih problema pri kreiranju ovog pisma.

Neke digitalne slovolivnice također kažu da je izuzalno laško konvartirati druge font formate u *Type 1* – prvenstveno zbog loge što se *Type 1* po svojoj podalkovnoj i algoritamskoj strukturi znatno razlikuje od svih drugih slovnih formata. Konačno tipografi su poneki put prisiljeni promijeniti ili u potpunosti izbeciti neka suptilna dijalova svojih kontura kako bi Adobeov rasterizator dao zadovoljavajuće rezultata.

Pa ipak, da budamo realni: *Type 1* daje odlične rezultate u velikom broju DTP aplikacija. Upravo te aplikacije na koje smo se već toliko navikli, kao i velik broj instaliranih *PostScript* sistema najveća su podrška *Type 1* koncepciji. Uz to, danas je dostupno na tisuća porodice *Type 1* fontova. Nikako nesmijemo zaboraviti i odlične programske alate (poput *Fontographera*) pomoću kojih možemo lako kreirati specifična YU karaktere, kao i potpuno nove *Type 1* fontove. Uostalom, *Type 1* je danas u toliko mjen rasprostranjen ne sistemima stolnog izdavanja, da nitko više niti ne razmišlja o njegovim maneme i eventualnom prelazu na novi fonl. Svi manje ili više, iraže načine da se domognu što većeg broja *Type 1* tontova. Zbog ovih razloga *Type 1* će se dosljedno vrijeme zadržati kao standerd, ne mareći za koncepcije i formate koji će pored njega prolazit.



4 Ovo bezserifno slovo ima dodijeljene tzv. nepostojeće trupove. Specificiranjem trupove gdja oni u stvari ne postoje font dizajneri prilijevaju *BuildChar* da i na ovakvim karakterima provede cjelokupan poetupak *peglanje*



3 Kod procese pegienje, Adobe *Type 1* fontovi primjenjuju tzv. trupova. Ti trupovi daju *BuildCher* resterizatoru globalne informacije o karakteru, kao što su širina trupova i serife, te gornji i donji rubovi slova

A SADA TRUE TYPE

TrueType razvijen od strene Applea, znatno se razlikuje od *Adobeovog Type 1 formata*



5 *TrueType*-ov sistem pegla obavlja svoju funkciju pomocu dveju vektora – projekcijskog i smjernog. Dok smjerni vektor specificira smjer u kojem se točka mora kretati, projekcijski vektor odreduje duljinu kojom će se točka pomicati.

Umjesto kreiranja još jedne više-razinske font paradigme Apple se odlučio na konstrukciju sistema pomocu skupa niskorazinskih primitivnih rutina kojima bi bilo moguće na jednostavan način implementirati sve ostale font formata uključujući i Adobeov *Type 1*. Drugim riječima konverzija ostalih fontova u *TrueType* format bila bi izuzetno lagana. Da bi široj javnosti lakše objasnio ovu mogucnost Apple je font formate usporedio sa programskim jezicima *TrueType* obzirom da je izgrađen na jednostavnim rutinama usporeden je sa nižim programskim jezicima – npr. asamblerom. *Type 1* zbog svojeg više-razinskog sustava predstavljao bi dakle više programska jezika – npr. paskal ili C. I sami znate da je lakše kompajlirati C ili paskal u asembler nego prevoditi C u paskal ili paskal u C.

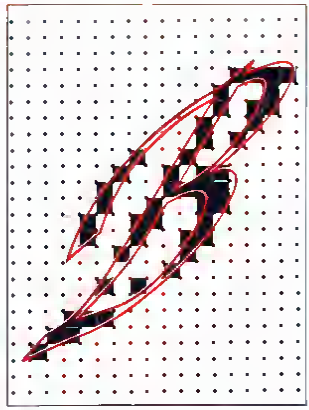
TrueType fontovi jednako kao i *Type 1* počinju sa konturama ali sličnost ovdje završava. Dok *Type 1* konsti programirane konture i deklarativno pegla*nje*. *TrueType* radi upravo suprotno. Konture nisu izrazene kao crtajuca instrukcija već kao skup kontrolnih tocaka. Te točke daju informacije *TrueType*-u koji na osnovu njihovog rasporeda i drugih podataka konstruira odgovarajuce krivulje. Spajanjem ovih krivulja formira se kontura karaktera. Ove su konture popracene zalim odgovarajucim kodom za pegla*nje*. Aktivirenjem ovog koda (koji kontrolne točke pomiče i raspoređuje tako da konture adekvatno pokrivaju raster) dobiva se finalni rezultat. Jednostavni sistem za rastezanjacuju sada lako moze kreirati bitmapirano slovo. Na ovome se temelji i globalna filozofija *TrueType*-a – uciniti model fleksibilnijim na laj nacin da se više inteligentnih funkcija ugradi u sistem za pegla*nje* a manje u rastezator.

TrueType-ov naglasak na manipulaciju konturama dovodi do neobicnog skupa instrukcija. Njegov stek orjentiran sistem sadrzi velik broj funkcija koje su niskorazinske u usporedbi sa drugim font formatima ali su izuzetno visoke ako ih usporedujemo sa konvencionalnim asemblerskim jezicima. Zato da bi shvatili ove funkcije potrebno je bolje poznavati nacin na koji *TrueType* manipulira kontrolnim točkama i sistem pomoću kojeg konture prilagodava rezoluciji rastera.

Pretpostavimo da dizajner zeli pomaknuti kontrolnu točku koja se nalazi na konturi karaktera i ima koordinate X Y na novu poziciju X1 Y1. Potsjetimo se da nemamo na raspolaganju apsolutni koordinatni sustav vec je cijeli sistem relativan. Drugim rjecima ne možemo dati nove koordinate na koje se točka treba pomaknuti u smislu X1 = A i Y1 = B. Obzirom da znamo da nam je jedina liksna točka upravo polazna točka, pomak točke možamo definirati kao X1 = X + n i Y1 = Y + m. U stvari ono što dizajner stvarno želi reći je „Želim da pomaknes ovu točku u tom smjeru ave dok njezin smjaštaj na poslane zadovoljavajuci."

TrueType omogucava i izvrsave ove pomake pomocu dvaju vektora – projekcijskog i smjernog. Dok smjerni veklor specificira smjer u kojem se točka mora kretati, projekcijski vektor određuje duljinu kojom će se tocka pomicati. Slika 5 prikazuje kako ovaj sistem radi. Točka A se pomiče duž projekcijskog vektora koji zatvara kut od oko 45 stupnjeva sa horizontalom. Duljina pomaka se mjeri na projekcijskom vektoru. (Projekcijski vektor je u ovom slučaju sasvim slučajno zbog lakšeg razumijevanja postavljen horizontalno, dok inače može biti postavljen i vertikalno ili u bilo kojoj ravnini.) Cijeli postupak potreban za pomak točke na ovaj način može se ostvariti pomoću jedne jedine *TrueType* instrukcije.

Pretpostavimo sada da želite premjestiti još jednu točku – ili cijeli skup točaka – zajedno sa prvom točkom, kako bi zadrzali originalnu geometrijsku formu. Ovo možete napraviti na laj način da kažete *TrueType*-u da je prva točka – uputna točka. Nakon toga potrebna je još samo jedna *TrueType* naredba za pomak jedne ili cijele grupe točaka zajedno sa prvom točkom. Dakle, ako želite cijelu desnu stranu slova pomaknuti gore desno kao na slici 5



6 Jednostavan kontrolni mehanizam pregledava samo vertikalne i horizontalne konture, popunjavajući praznine samo na tim mjestima. Kod karaktera poput ovog na slici, koji imaju duge, zakrivljene dijelove *TrueType*-ov mehanizam nece radrti

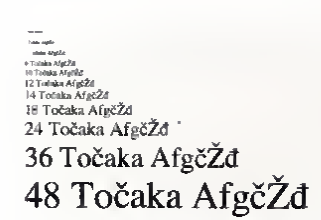
možete postupak sa te slike primjeniti na sve točke u tom dijelu slovna kontura.

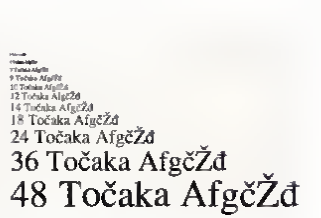
Jedna druga samostalna instrukcija pronalazi mjesta na kojima se ukrštavaju linije (ova se sjecišta nalaze na mjestima gdje su krajnje linijske točke) i ta područja ispunjava konturama. Postoji takodar i funkcija kojom *TrueType* prilagodava kuteve medu linijama kako bi cjelokupna kontura bila što ispeglanija. Obzirom da se radeći sa velikim brojem kontrolnih točaka možete izgubiti ne znajuci više koje ste tocka odabrali. *TrueType* posjeduje *loop* varijablu koja sadrzi podatke o broju točaka. Pozivom ove varijable možete u svakom trenutku saznati na koliko se točaka odnosi instrukcija koju cete primjeniti što vjerujta može biti izuzetno korisno.

TrueType sadrzi i nekoliko mehanizama za manipulaciju karaktera na niskim rezolucijama. Kontrolna tablica iz ovog mehanizma može se koristiti za prsirjavanje osjetljivih dijelova karaktera poput trupova i serila da se širinom prilagode maloj veličini ili u izuzetnim situacijama ako nema dovoljno prostora da se ti dijelovi jednostavno izbrisu. Konacno kada radite u *TrueType*-u sa slovima izuzetno malih veličina a rezultat koji vam nudi rasterizator vas ne zadovoljava, možete jednostavno odustati od rasterizacije i konstiti funkcije za iscrtavanje bitmape. Medutim ovo krajnje rješenje ne predstavlja najidealniju soluciju. Za definiranje jednog malog bit mapiranog slova potrebno je napisati izuzetno puno instrukcijskog koda što nes izmedu ostalog vraća i na primarne probleme bitmapiranih fontova.

Vjerujem da vas do sada čitajuci gornje redove, *TrueType* nije nili najmanje podsjecao na neki „asemblerski jezik". Pa ipak sakrivene i dobro zepakirane u *TrueType*-ovim solisticiranim grafičkim instrukcijama nalaze se jednostavne ili bolje rečeno primitivne naredbe (kao JUMP ili CALL) koje možete susresti (da samo susresti) i u bilo kojem drugom nižem programskom jeziku.

Za razliku od *Type 1* font programa *TrueType* programi imaju pristup velikom broju ma





7 Na lijevoj strani prikazan je Times-Roman font u *Type 1* formatu ispisan na iseerskom pisaču razlučivosti 300 tpi. Cijene ispisne jedinice $4822, cijena iste ispisne jedinice sa *TrueType*-om $2268. Desno isti font u *TrueType* formatu ispisan na tintnom pisacu u rezolucIji od 360 tpi. Cijena ispisne jedinice $890.

lemaličkih operacija zbrajanju oduzimanju množenju dijeljanju negaciji zaokruživanju i Bulovim operacijama To ujedno znači da *TrueType* program može vršiti kalkulacija ve zane uz bilo koji drugi font sistem

Ukratko *TrueType* u biti nije jedan jezik već dva osnovni stek-orijentiran jezik i skupina specijalnih visoko-razinskih rutina za manipulaciju konturama

U *TrueType*-ovoj dokumentaciji i specifikacijama (u kojima je sadržano i objašnjeno 112 jednobajtnih instrukcija i fajl formata) nija posvecena veca paznja na objašnjanje fundamentalne filozofije *TrueType* nili je prikazano na koji su način prikazane instrukcije iskorištene u programu Pa ipak rijetko ce koji konsnik imali potrebu da ulazi u sam bit *TrueType*-a i da se nepotrebno zamara sa kompliciranom jezičnom strukturom Ovo je i najvjerovatniji razlog zbog kojeg su la objašnjenja i izoslavljena Trenutno vodece fonl-softverske kompanija – uključujući Bitstream URW Allsys i Monotype – rade na razvoju programskih alata koji ce automatski „kompajlirati" fontove u *TrueType* format Na taj način kao i programeri RISC sistema većina fonl dizajnera neće se nikada susresti sa instrukcijama na kojima su izgrađani *TrueType* fontovi

POZITIVNO – NEGATIVNO

TrueType uprkos odličnim kritikama od strane renomiranih tipografskih i računarskih stručnjaka ipak ima nekoliko nedostatake Prvo on proizvodi znatno veće datoteke od vecina drugih formata uključujući i Adobeov *Type 1*. Ie Bitstreamov *Speedo* Drugo kontrolni mehanizam zahtijeva mala poboljšanja Naime u lehničkoj dokumentaciji stoji da je *TrueType* u mogućnosti skanirati samo honzontalne i vertikalne konture Ie popunjavati praznine samo na I.m mjestima. dijagonalne linije ostaju dakle nesređene (pogledaje sliku 6) Ipak važno je napomenuti da se ovaj nedostatak u normal noj svakodnevnoj praktičnoj primjeni uopće ne može primjetiti

Kada govorimo o manama *TrueType*-a nikako ne smijemo zaboraviti i činjenicu da vacina današnjih DTP aplikacija u prvom redu podržava *PostScript* i njegov *Type 1*. što nikako ne ide u prilog *TrueType*-u Iskustva nam lekoder govore da svaki proizvod pokazuje prave dobre i loše strane upravo na tržištu medu korisnicima Adobeov *Type 1* je svoje tržišno kraljenje dobivo odavno suprotno toma *TrueType* se na tom području Jek Ireba dokazati

Apple-Microsoftova fonl koncepcija u svakom slučaju ima i niz pozitivnih strana Proizvodači ugradnju *PostScript* interpreter zajedno sa *Type 1* programima, u hardver, točnije u ROM ispisnih jedinica Ukoliko dakle zelite svoju staru verziju *PostScript*-a zamijeniti novom, preostaje vam jedino da kupite novu ispisnu jedinicu, ili da krenete u pohrapu za ROM om sa novijom verzijom *PostScript* a (ova posljednja završila bi negdje na Jupiteru) Sa druge strane promolon *TrueType*-a misla ovaj format implementirali u operativni sistem svojih računala dakle u softver a ne u hardver Na ovaj ca način konsnici daleko lakše pratili tehnološka poboljšanja bez da kupuju novu opremu

Kao sljedeću *TrueType*-ovu prednost mogu navesti njegovu brzinu On se naime pri ispisu pokazao daleko brži od Adobeovog *Type 1* koda Uz to sam *TrueType* odnosno njegov programski kod je izuzetno malen Ie ga je lako ugraditi u aplikacije ili čak translirati u *PostScript* i zatim lakav kod slati na *PostScript* pisač Obzirom da podrzava dvo-bajtni karakter sel i nelinearno skaliranje *TrueType* predstavlja odlicnu podršku za ideografske fontove

Konacno dok Adobe mora sakrivati odredene dalalje svoje koncepcije od sire javnosti da bi se odrzao u poslu, Microsoft i Apple su ponudili brzišlu kompletnu *TrueType*-ovu lehnicku dokumentaciju Uz to Apple i Microsoft namjeravaju *TrueType* dijalili besplatno Iočnije kao dio svojih operativnih sistema Na taj je način Apple već ugradio *TrueType* u svoj novi sistem Mac OS 7 0 a Microsoft najmjerava uciniti isto u Windowsima 3 1 Ie u OS/2 *Presentation Manageru* (Činjenica da fontovi i jezici za opis stranice nisu primarni produkti ovih firmi daje im znatne prednosti nad Adobeom)

Iako do današnjeg dana Microsoft nije isporučio nili jedan *TrueType* produkl Apple je sa svojim *TrueType* ispisnim jedinicama na trzišlu već više od 3 mjesaca. Negdje oko maja Apple je javnosti predstavio svoja dva nova visoko razlučiva pisača sa naglaseno niskim cijenama od kojih je jedan baziran na Canonovom laserskom sistemu a drugi na Canonovom Bubble Jet mehanizmu Dakle dok je prvi laserski standardna rezolucije 300 Ipi, drugi je tinlni rezolucije 360 Ipi Zajedničko za oba pisaca je upravo to da nili jedan od njih na podrzava *PostScript* kao jezik za opis stranice, već oba korista Macintoshev *QuickDraw* mehanizam koji je uoslalom već odavno dio Macintosheovog operativnog sistema Naravno Apple uz oba printera isporučuje i pakel *TrueType* fontova koji se sastoji od Helvetice Timesa Courierа i Symbola Važno je lakodar napomenuti i to da Apple za jugoslavensko tržište prako svojih Apple Centara i ostalih dilera isporučuje ove fontove sa YU-znacima (pogledajte sliku 7)

BUDUCNOST

U idealnom svijetu fonl dizajneri bi kupovali programske alate pomoću kojih b kreirali kompletne perfektno ispeglane fontova na kojima konsnik ne bi morao vrsiti nikakve naknadna intervencije Korisnici bi zatim kopija lakvih fontova kupovali u obliku univerzalnih formata koji bi se mogli konstiti na bilo kojem racunalu neovisno o operativnom sistemu aplikacij ili proizvodaču

Iako smo danas daleko od ovog idealizma Adobe je sa svojim *PostScript* om i *Type 1* formalom gotovo zaokružio Iu ideju Pojava *TrueType*-a nanovo je uzdrmala usposlavljene standarde, stvarajući probleme prvensivno korisnicima koji sada ne znaju za koju da se koncepciju opredijale Pa ipak ova će se situacija ubrzo srediti U sljedećim mjesecima Microsoft će izbaciti svoje operativne sisteme (*Windows 3 1* Ia OS/2) sa ugrađenim *TrueType* fontlovima. a u dogledno vrijeme možemo očekivati da obje kompanija (Apple i Microsoft) počnu ugradivati i Microsoftov *TrueImage* je zlk za opis stranice koji bi sa svojim perfor mansama u potpunosti mogao zamijeniti *PostScript* Pa ipak moramo biti svjesni da će se *PostScript* održati još dugi niz god-na ali u lim će ga godinama polako ali sigurno smjenjivati Apple Microsoftova TRUE koncepcija

Korisna adresa

DTP Studio NOVEL
41000 Zagreb Milane Rešetara 44
Tal 041/ 34 75 14 Fax 041/ 34 79 43

Drajveri

Nth Drive
Display List softver za *AutoCAD*

AutoCAD sam po sebi podržava nekoli ko najpopularnijih grafičkih adaptera za PC ponajpre one stvorene u IBM Vecina ostalih grafičkih karti koje još nisu zvanič no standard a ima ih svuda okolo kao Tseng ET 4000 ili TIGA na primer moraju da se oslone na dodelne drajvere Najnovija generacija tih drajvera su „display list" drajven Kod kartica sa grafičkim procesonma vektorska lista crteže smešla se u Display List RAM grafičkog procesora i on odatle crta model brzinom obično znatno vecom nego što bi CPU ikada mogac, pa i kada je u pitanju 50 MHz 486

Odnedavno se uz neke grafičke karte kao nejnoviju VESA + CEG + HiColor ET4000 VGA karti Iakodje daju „display list" drajven Kod njih se formira ekranska lista ali je ona u RAM (naravno u zaštićenom načinu) i crtanje tu vrši procesor Pomoću ove ekranske liste se dobija u brzini tri operacije **Redraw Pan** i **Zoom** Neko će reći samo Ie tri operacija ali one čine najveći deo operacija koje se konsle lokom rada se **AutoCAD** Ekranska lista omogućuje i da se vrši **Zoom Ali** (odnosno do 99% veličine crteža) bez ponovnog mučnog regenerisanja U svekom slučaju lepo je kada se ovakav drajver dobija uz kartu

Nth Graphics poznati proizvodjač Nth Engine grafičkih kartice odlučio sa da napravi svoju verziju Display Lisl drajvera za (skoro) sva VGA I Nth 8514 karte Nth Dnve, kako je nazvan vrši sve tunkcije koje vrše i drajven koc novih Tseng ET 4000 karti s tim što ima izbor veličina i stilova slove za meni I komandni red *AutoCAD*-a 11 386 Drugi dodatak su „Birdseye" i „Spy glass" načini kombinovanog zumiranja i panirenja, koji su već klasika kod svih drajvera za kartice sa grafičkim procesonma I dok izbor stilova slove u desnom meniju i komandnom redu i nije baš neka prednost (on) ionako ustupaju mesto padajućim menijima) **Zoomeldes** u svakom slučaju jesu Ti slajdovi se razlikuju od klasičnih *AutoCAD* slajdova u tome što se sa njima mogu vršiti **Zoom** I **Pan** operacije

Zbog veoma kratkog vremena koje nam je bilo dato, mi nismo uspeli da pregledemo sve mogućnosti ovog drajvera na redakcij skoj VGA karti ali po prvim utiscima možemo reći da brzinski dobtak i nije baš neki – rede 20 – 60% u Iri osnovne operacije Taj dobitak je uz to jedno moguć pod uslovom da imate preko 8 Mb RAM a ko već daje tolki ulog, može da nabavi i neku 8514 Ili TIGA kartu i za ispod 1000 USD uz koju će mu rad u svakom slučaju biti brži

Najlepši deo Nth drajvera je u svakom slučaju njegov HARD LOCK – da dobro ste pročitali Drajver od par zelenih stotki sa hard lockom Sve u svemu, Nth Dnve donosi neke prednosti konsniku **AutoCAD**-a ali nije vredan nekih dodelnih para Uz kartice se već besplatno daju slični drajveri a eko računamo par dodatnih funkcija realna cena Nth Dnve bi bila oko 50 USD Po toj ceni (ali nikako većoj) vredi ga kupiti

Konlakt adresa. Korisne adresa ***ABV Graphics****, 11000 Beograd, Andričev venac 4 tel/fax 011/345-277*

N Novaković

VIŠE OD EGZOTIKE

Reklo bi se da su C i paskal definitivno okupirali računare programera širom sveta, prognavši sa tvrdih diskova sve ostale programske jezike pravo u istoriju. Stvari, međutim, nisu baš tako jednostavne. Mnogi programski jezici, iako ih često smatraju čistom egzotikom, i dalje se dobro drže. Među njima je i modula-2.

Duško Savić

U početku beše Logitec *Modula-2*. Njena pojava u ovim krajevima izazvala je pravo zaprepašćenje umesto idealnog prevodioca, kako je to po računarskim časopisima Zapada bilo opisivano, dobili smo praktično neupotrebljiv proizvod po svemu nalik na MS FORTRAN iz prvih dana XT računara. Prevođenje je trajalo „danima", uz stalno konsultovanje diska pa je svaki programer koji drži do sebe i svog vremena odmahnuo rukom i promrmljao našto-poput „*Turbo Pascal* je bolji." Na sreću, svaki Microsoft ima svog Borlanda kao konkurencija Logitecovoj modul 2 pojavila se *JPI Modula 2* pravljena po ugledu na *Turbo Pascal* razuman editor, prevođenje iz mamonje (koliko god je to moguće u ovom jeziku) velika brzina izvršavanja pravedenog programa. Ljubitelji modula-2 su odahnuli i prionuli na posao budno motreći novitete koje ja firma JPI redovno nudila.

Tokom minule dva godina *JPI Modula-2* postala je praktični standard za modulu-2 na PC računarima. Početkom ove godine pojavio se prevodilac izmenjenog imena i još boljih karakteristika – *TopSpeed Modula-2*. Ime je novo, jer firma JPI pod isti kišobran stavlja prevodioce za nekoliko najmodernijih jezika, kao što su ada, C, C++ i modula-2 i paskal.

VIŠEJEZIČNI KONCEPT

Firma JPI posvetila se isključivo izradi i plasmanu prevodilaca za navedene jezike. Prvi proizvod joj je bio *JPI Modula 2*, verovatno zato što je prevodilac za modulu manji nego za paskal pa ga je lakše napraviti. Takođe konkurencija je tu mnogo manja nego za paskal i drastično manja nego za C. Naravno, jedan prevodilac nije dovoljan i u firmi JPI su se dosetili da dobre strane svog prevodioca za modulu-2 prenesu i u druge jezike. I ne samo to JPI je sebi postavio za cilj da na PC računarima ostvari tzv. višejezično programiranje.

Čest je slučaj da programer radi na jednom jeziku a da mu treba nešto što već poseduje u nekom drugom jeziku. U toj situaciji su najčešće programeri koji ne pišu na C-u, jer za taj jezik postoji ogromna ponuda gotovih potprograma. Naivni pokušaji da se iz paskala koriste rutine za C (ili obratno, svejedno) propadaju pri prvom pokušaju. U jezici su interno isuviše različiti pravili su ih programeri vođeni sasvim različitim filozofijama. Ukratko, od kompatibilnosti ni traga ni glasa. Koncept višejezičnog programiranja zasniva se upravo na ispunjavanju takve jedne potrebe. Kada bi takav koncept zaživeo u praksi, bilo bi realno pisati glavni program na modulu-2, podacima u npr. dBASE formatu pristupati iz C-a, ispis teksta na ekran rešiti nekom C++ biblioteko i slično. Ne samo da bi jedan programer koji zna više jezika mogao da efikasno konsti programske resurse, nego bi bilo omogućena bezbolna simbioza rada čitava grupa programera od kojih je jedan specijalista za modulu-2, drugi za paskal, treći za C itd.

Ovakav koncept za sada ima samo firma JPI. Realizacija je mnogo lakša nego što bi se moglo reći na prvi pogled. Svaki jezik treba da ima editor i radnu sredinu – zašto svi ne bi imali isti? Svaki jezik treba da ima generator koda za procesore PC računara – zašto svi ne bi imali isti? I tako dalje: jedina prava razlika je u sintaksnom analizatoru. U porodici *TopSpeed* jezika sintaksni analizator za svaki pojedinačni jezik zauzima oko 100K, a realizovan je kao prepokrivanje (overlay) editora. Prosto učitavanjem prepokrivanja dobija se potpuno novi jezik. Idealno bi bilo da se prepokrivanje poziva iz u osnovi istog ali takoda i dovoljno programabilnog editora, na primer iz editora či je bi naredbe bile u ASCII datoteci na disku.

Verovatno na treba posebno naglašavati da je firma JPI uspela dosledno da sprovede to što je naumila. Evo kako to izgleda na primeru *modula-2* verzija 2.0.

PAKET I INSTALACIJA

Paket se sastoji iz dva osnovne priručnika, koji se zovu *Language Tutorial* (210 strana) i *User' Manual* (DOS version, 410 strana). Kao dodatak osnovnom paketu može se dokupiti *TechKit* sa priručnikom od 144 strane. Priručnici su dobro opremljeni a pisani su razumljivo i jasno. Instalacija se odvija jednostavno, pokretanjem odgovarajuceg programa INSTALL za samu *Modulu 2* odnosno za *TechKit Modula 2* zauzima ih a *TechKit* zauzima dva i po megabajta na tvrdom disku, instelacija na diskete bila bi praktično nemoguća. Po završenoj instalaciji na disku ostaje imenik (diradory) TS sa deval podimenika. Kada bi na tom disku bio instaliran još neki prevodilac iz *TopSpeed* serije i on bi bio smešten u niz podimenike „glavnog" imenika TS. Sve je, dakle, podređeno ideji o višejezičnom programiranju.

EDITOR

Naredba TS iz DOS-a aktivira editor odnosno radnu sredinu za *TopSpeed* jezike. Rad sa tekstom rešen je po ugledu na *Wordstar*. Bitna novost u radu sa blokovima je naredba Ctrl KU kojom se jednom poništeni blok moža „spasiti od zaborava" i premeštati proizvoljno po tekstu. Prozora ima čak deset, ali je jedan rezervisan za sam sistem, tako da se datoteke mogu posmatrati istovremeno u devet posebnih prozora. Podrazumeva se da korisnik može podesiti atribute svakog prozora pojedinačno, boju osnove i okvira, položaj i veličinu itd. Ugrađena je i podrška za EGA i VGA rezolucije pa je sasvim lako gledati definicione i izvršne module zajedno. Prava vrednost ovakvog editora vidi se tek u višejezičkom radu, jer u dva-tri prozora može biti program u modulu a sledeća dva-tri prozora objekti u C++ u, i u preostala dva-tri prozora možemo razgledati programa pisane u *TopSpeed Pascal*-u.

Miš lakoda radi a makronaredbe posluje i upotrebljiva su. Makronaredbe se zapravo dale na dva grupe: na jednoslovne i dvoslovne. Jednoslovne se aktiviraju pritiskom jednog tastera u kombinaciji sa tasterom Alt, dok se dvoslovne aktiviraju iz dva koraka: prvo zajednickim pritiskom na tastere Alt i minus - pa tek onda na samo slovo. Kao i u svim profesionalnim makrosistemima možemo snimati makronaredbe interaktivno ili ih pisemo u posebnom ASCII editoru. Posluje i posebne naredbe koje se sa tastature na mogu proizvesti. Uglavnom se odnose na protok vremena, odnosno zaustavljanje izvršenja makronaredbe do nekog događaja, npr. dok neki taster ne buda pritisnut i slično. Ako konstimo te dodatna mogućnosti, moramo pisati makronaredbe kao ASCII datoteke i pravoditi ih posebnim programom koji direktno menja sadržaj editora.

Makronaredbe u *TopSpeed* editoru rešene su profesionalno, ali nema smisla konstbi ih za totalnu promenu interfejsa. Time bismo izgubili vezu sa dokumentacijom, a poslojeći sistem menija i naredbi ionako ne predstavlja nikakvu smetnju za svakodnevni višečasovni rad.

Opcije glavnog menija izgledaju sasvim obično: **Files** (datoteke), **Edit** (izmena teksta), **Compile** (prevođenje), **Make** (kontrola ispravnosti varzija), **Run** (izvršenje programa), **Project** (kontrola nad svim datotekama iz kojih sa pravi program), **Options** (opcije), **Utilities** (pomoćni programi) i **VID** (dibagar). U odnosu na verzlju 1. primacujemo da je opcija **Link** zamenjena opcijom **Project**, koja se u slučaju korišćenja samo jednog *TopSpeed* jezika svodi na **Link**, ali može i mnogo više, ako zaisla radimo sa dva ili više jezika istovremeno. **Project** će kod prevođenja pozivati prevodilac za modulu ako je teksi na moduli prevodilac za C++ ako je taksi programa na C++-u itd.

Opcija **Make** je rešena slično istomenoj opciji u *Turbo* editorima: ako se prevodi iz editora korisnik ništa ne mora da zna o redosledu prevođenja programskih modula, a za slučaj da se žele naznačiti komplikovanije veze između programskih delova, moguće je zadavati naredbe za **Make** i iz posebne ASCII datoteke. Slično, i za opciju *Project* postoji posebna ASCII datoteka sa odgovarajućim naredbama. Za *Project* postoji posaban programski jezik a služi za definisanja najvažnijih parametara kao što su vrsta prevedene datoteke, način povezivanja programa sa posljećim sistemskim datotekama i memorijski model. *Project* može generisati sledeće vrste prevedanih programa: DOS EXE, OS2 EXE, DOS DLL, OS2 DLL, Library, DynaEXE, WinEXE i WinDLL. Memorijski modeli su tipa **Small**, **Compact**, **Medium**, **Large**, **XLarge** i **MThread** a osim njih konsnik može definisati i sopstveni memorijski modal.

Osim rada uz pomoć editora programa možemo prevoditi iz DOS-a pozivom alternativnog prevodioca po imenu TSC. Sva opcije koje možemo postaviti iz editora možemo lakoda postaviti i iz TSC-a, a konsilćamo ga za pisanje programa i podataka koji zahtevaju celokupnu DOS memoriju za sebe.

TopSpeed Modula 2 može da prevede program za različite operativne sistema i radna sredina. Sva izdanja mogu da naprave DOS

Listing 1

```
(* Primer sa uslovno prevodjenje *)
MODULE Hello2;
IMPORT IO;
CONST Francuski = FALSE;
      Engleski  = TRUE;
BEGIN
(*$T Francuski *)
    IO. WrStr("Bonjvar");
(*$E*)
(*$T Engleski *)
    IO.WrStr("Good day");
(*$E *)
END Hello2.
```

Primer uslovnog prevodenja

Listing 2a *JPI Modula-2 v2*

```
(* JPI Modula-2 v2 *)
MODULE MERENJEi;
VAR i, j, k : CARDINAL;
    r : REAL;
BEGIN
FOR j := 1 TO 100 DO
      FOR i := i TO 65000 DO
        r := r + 0.0;
        END;
    END;
END MERENJEi.

// Turbo C++ v2.0
#include   <stdio.h>
#include   <dos.h>
int main (void) // MERENJEi;
{
```

Listing 2b *Turbo C++ v2.0*

```
unsigned int i, j, k;
float r;
   struct  time t;
   gettime(&t);
   printf("Vreme je: %2d:%02d:%02d.%02d\n",
          t.ti_hour, t.ti_min, t.ti_sec, t.ti_hund);
   for (j=1; j <= 100; j ++)
     for (i=1; i <= 65000; i ++)
        r = r + 0.0;
   gettime(&t);
   printf("Vreme je: %2d:%02d:%02d.%02d\n",
          t.ti_hour, t.ti_min, t.ti_sec, t.ti_hund);
   return 0;
}
```

Listing 2c *Turbo Pascal 6.0*

```
{ Turbo Pascal 6.0 }
PROGRAM MERENJE1;
uses DOS;
VAR i, j, k : word;
    r : REAL;
    h, m, s, hund : Word;
function LeadingZero(w : Word) : String;
var s : String;
begin
  Str(w:0,s);
  if Length(s) = 1 then s := '0' + s;
  LeadingZero := s;
end;
begin
GetTime(h,m,s,hund);
WriteLn('Vreme je: ',LeadingZero(h), ':',
          LeadingZero(m),':',LeadingZero(s),
          '.',LeadingZero(hund));
FOR j := 1 TO 100 DO BEGIN
    FOR i := 1 TO 65000 DO BEGIN
        r := r + 0.0;
        END;
    END;
GetTime(h,m,s,hund);
WriteLn('Vreme je: ',LeadingZero(h),':',
          LeadingZero(m),':',LeadingZero(s),
          '.',LeadingZero(hund));
END.
```

Isti programerski zadatak, tri popularna programska jezika, drastične razlike u efikasnosti

Listing 3

```
DEFINITION MODULE Tacka0;
TYPE Tacka = CLASS
        x, y : INTEGER;
        PROCEDURE NovaTacka(xi, yi : INTEGER);
        PROCEDURE DuzinaTacka() : LONGREAL;
        END;
END Tacka0.

IMPLEMENTATION MODULE Tacka0;
FROM MATHLIB IMPORT Sqrt;
TYPE Tacka = CLASS
        x, y : INTEGER;
        PROCEDURE NovaTacka(x1, y : INTEGER);
        BEGIN
            x := xi;
            SELF.y := y;
        END NovaTacka;

        PROCEDURE DuzinaTacka() : LONGREAL;
        BEGIN
            RETURN Sqrt(VAL(LONGREAL, x * x + y * y));
        END DuzinaTacka;
    END; (* CLASS *)
END Tacka0.

DEFINITION MODULE Tacka3;
FROM Tacka0 IMPORT Tacka;
TYPE Tacka3D = CLASS(Tacka)
        z : INTEGER;
        PROCEDURE NovaTacka(x1, y, z : INTEGER);
        PROCEDURE DuzinaTacka() : LONGREAL;
    END; (* CLASS *)
END Tacka3.

IMPLEMENTATION MODULE Tacka3;
FROM MATHLIB IMPORT Sqrt;
TYPE Tacka3D = CLASS(Tacka)
        z : INTEGER;
        PROCEDURE NovaTacka(x1, y, z : INTEGER);
        BEGIN
            Tacka.NovaTacka(x1, y);
            SELF.z := z;
        END NovaTacka;

        PROCEDURE DuzinaTacka() : LONGREAL;
        BEGIN
            RETURN Sqrt(VAL(LONGREAL, x*x + y*y + z*z));
        END DuzinaTacka;
    END;
END Tacka3.

MODULE Tacka0;
IMPORT Tacka0, Tacka3, IO;
VAR t2 : Tacka0.Tacka;
    t3 : Tacka3.Tacka3D;
BEGIN
    t2.NovaTacka(3, 0);
    IO.WrLngReal(t2.DuzinaTacka(), 0, 14);
    t3.NovaTacka(2,3,0);
    IO.WrLngReal(t3.DuzinaTacka(), 10, 4);
END Tacka0.
```

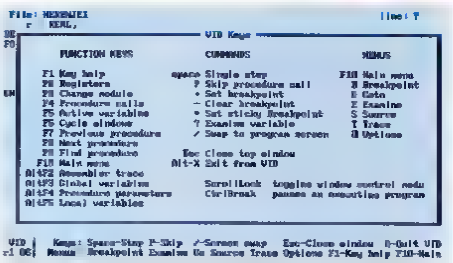
Primer za objektno programiranje

EXE i **Library**. Ti da prevedu program za DOS ili za sistemsku biblioteku. Za OS2 verzije (OS2 EXE i OS2 DLL) treba kupiti posebnu verziju prevodioca. Za *Windows* (WinEXE i WinDLL) i dinamičke verzije (DOS DLL i DynaEXE) treba dokupiti *TechKit*. Pri tome dodajmo i da ova verzija module-2, 2.0 zahteva prisustvo Mikrosoftovog *SDK for Windows* da bi program tipa WinEXE zaista mogao da se prevede tako da se izvršava pod *Windowsima 3.0*

MEMORIJSKI MODELI

Procesori 8088 i 8086 mogu da adresiraju ceo megabajt memorije (šinne prostupe 20 bita) ali su im registri najviše 16-bitni potrebna su dva registra da bi se mogle zadati apsolutna adresa u celom adresnom prostoru. Veća prostupa memoriji odigrava se ipak serijski pa za maksimalno adresiranje uglavnom nema potrebe. Zato je intel smislio segmentiranje memorije tj. podelio je memoriju na segmente maksimalno dugačke do 64 kilobajta jer jedan 16-bitni registar upravo toliko memorije može da adresira sam za sebe. Posledica segmentirane memorije je da nijedan programski jezik na početku PC računarstva nije mogao da od DOS-a zatraži i dobije niz duži od 64 kilobajta.

Segmentirana arhitektura je i danas ozbiljna prepreka pisanju većih programa jer programer nikada ne vidi memoriju kao nepreskidan niz bajtova nego isključivo kao niz od najviše 64 kilobajta segmenata. Adresa u okviru segmenta su dvobajtne (near pointers) a sve adrese van jednog segmenta zahtevaju četiri bajta (far pointers). Adresa aktuelnog segmenta se obično nalazi u registrima CS ili DS, u CS je početak segmenta sa izvršnim naredbama a u DS



je početak segmenta sa podacima. Smisao uvođenja memorijskih modela je osiguravanje da ova dva registra uvek sadrže ispravne vrednosti

TopSpeed Module 2 ima šest već navedenih memorijskih modela. Takvi standardni modeli nalaze se i u drugim programskim jezicima namenjeno u prevodiocima za C. Nov je jedino šesti model. **Multithread** kojeg treba koristiti za opisivanje procesa ili za rad pod OS2. Ovaj memorijski model se takođe preporučuje za konvertovanje programa iz prethodne verzije *Module-2*.

Za svaki memorijski model postoji odgovarajuća biblioteka na disku. Kod instalacije možemo odbiriti prenos samo onih biblioteka čije ćemo memorijske modele stvarno i koristiti.

Interesantno je da možemo i sami definisati nove memorijske modele. Memorijski model je skup od sedam parametara: **far calls** (daleki pozivi) **large data** (podaci veći od jednog segmenta) **double word pointers** (četvorobajtni pokazivači) **global threshold** (maksimalna dužina podataka ako je podatak duži biva smešten u poseban segment) **JPI calling convention** (vrsta poziva) **Multi-Thread** (da li je biblioteka „re-entrant" ili nije, tj. da li sa naznim meštima u memoriji može upotrebiti ista osnovna verzija neke biblioteke potprograma) **Library ID** (ime odnosno, identifikacija biblioteka na disku). Svaki od ovih parametara korisnik može poslaviti i sam i tako stvoriti novi model. Primera radi, ako su programu potrebni samo po segmenti za naredbe i segmenti za podatke treba odabrati **Smell** model. Ali ako je pri tom potreban i pristup apsolutnoj adresi van segmenta potreban nam je dalekl pokazivač (**far pointer**). Možemo ga uključiti biranjem **Compact** modela, ali će najelikasnje biti da sami dalinisamo svoj memorijski modal (u **Smell** modelu samo poslavimo odgovarujuću opciju na on). Rezultat je da je pristup programu i podacima ostao brz (za to sa koriste **near pointers**), a da pomoću far pointera možemo doseći neki drugi segment (tabela 1)

NOVI POKAZIVAČI

Novi memorijski modali se obično prave na osnovu **Small** modela, ali možemo ih napraviti končenjem odgovarajućih pragmi ili direktnim konscanjem naredbi **NearADDRESS** i **FarADDRESS**. Naime *TopSpeed Modula 2* definiše tri generička tipa pokazivača

```
VAR
   x : ADDRESS;
   y : NearADDRESS;
   z : FarADDRESS;
```

Pokazivač **y** će biti uvek dvobajtni bez obzira na aktuelni memorijski modal. Slično z će uvek biti dužine šesin bajta, tj. služiće kao **far pointer**. Tip pokazivača **x** zavisiće od memorijskog modala ili od akluelnih pragmi

PRAGME

TS editor je zamišljen kao bogato opremljena radna sredina što znači da postoji mnogo načina da se programerove želje ulkaju u sistem. Najednostavniji je, opcijom **Options** glavnog menija odnosno. ulaznim parametrima za TSC. Ta dva nacina nisu uvek i najbolja možda zaboravimo tačnu konfiguraciju parametara ili nam možda nija lako reprodukovati ili kucati parametre. Zato postoji i treći način u *Project* dalolect svakom delu programa možemo postaviti posebne parametre, zahvaljujući postojanju posebna službene reči **override**. To je llaksibino ali nije očigledno. Najbolji način je da upulstva prevodiocu budu zapisana u samom lakslu programa. Obično se za to koriste tzv direktive npr. komentari sa posebnom sintaksom. Prethodne verzije *Module-2* koristila su upravo tu vrstu zapisa. Tako je komentar

```
(*$I+*)
```

oznacavao proveru indeksa i to je sasvim nekako za programera (prevodiocu je ionako svejedno pod uslovom da je sintaksa jednoznačne). U skladu sa proklamovanim konceptom višejezičnog programiranja, firma JPI je osetila potrebu da standardizuja poruke prevodiocima. Od svih jezika koje JPI podržava jedino ada ima definisanu sintaksu za direktive i odatle je preuzalo i ime (pragma) i sintaksa. Naravno, neće svaka pragma vaziti u svakom pojedinačnom jeziku, ali će bar sintaksa ostati ista

Pragma u moduli 2 je i dalje poseban oblik komentara, koji počinje sa **(*** Tipična pragme bile bi npr

```
(** call( o_a_size => on) *)
(** data ( var_enum_size => off ) *)
(** call ( c_conv => off, near_call => off
) *)
```

Uzgred poslednja pragma u C-u bila bi napisana ovako

```
#pragma call ( c_conv => off, near_call
=> off )
```

jer je u C-u **#pragma** definisano

Pragmama se može poslaviti sve što i opcijama iz menija ali i više. Ta dodatna mogucnost licu se poziva procedura rasporeda podataka po segmentima, snimanju pragmi na disk, javnih imena i provera koje treba umetnuti u prevedeni program

USLOVNO PREVOĐENJE

Posloji i uslovno prevođenje u skladu sa slanjem nake konstanta prevodiča se različili delovi istog programa. Uslovno prevodenje najvecu primenu nalazi tokom testiranja programa ili kad treba napraviti nakoliko različitih verzija istog programa (demo verzije). U priloženom programu (listing 1) prevasce se samo red IO.WrStr("Good day") jer je vrednost konslante **engleski** poslavljena na TRUE. Osim konstanti koje programer sam postavi posloje i predefinisane konstante za „letece" izmene memorijskih modela

OPTIMIZACIJA

Nivoom optimizacije u *TopSpeed Moduli 2* upravlja se preko jedanaest parametara, npr. za optimizaciju vremene izvršenja programa naspram veličine izuzimanje zajednickih izraza i konstanti, optimizacija preskoka petlji konšćenje registara i slično. Moguca je generisali kod posebno za svaki od procesora 8086, 80286, 80386 i 80486. I za svaki od koprocesora 8087, 80287 i 80287. Naravno ako koprocesora nema, biće emuliran softvarski

Koliki je efekat ovih optimizacija? Priložen je isli program na tri jezika – *Modula 2* V2, *Turbo C++* i *Turbo Pascal 6.0*. Vremena izvršavanja tih programa bila su 3, 23 i 34 sekunde. Na *Moduli 2* ali u verziji 1.15 isti program se izvršavao 24 sekunde dakle uporedivo sa *Turod* C++. Za ljubitelje *Winodwsa 3.0* program prevadan sa M2 V2 izvršavao se 14 sekundi – mullitasking košta! Zauzvrat editor raspoznaje prosirenu (**expanded**) memoriju i radi iz nje lako da je proces stvaranja programa pod *Winodwsima* (jer oni automatski otvaraju i *expanded* i *extended* memoriju) još brzi nago lnače

Predloženi program je indikativan jer svaki realan program ima bar jednu petlju i u njoj sabiranje realnih brojeva. Naravno bilo bi praurarijeno reci da je *JPI Modula-2* „najbrža na svetu" ali razlike u brzinama zaista su dramatične

Tabela 1 Memorijski modeli

vrata	sraall	compact	mediura	large	XLarge	MT
far calls	off	off	on	on	on	on
large data	off	off	off	on	on	on
double word pointers	off	on	off	on	on	on
global threshold	64K	32K	64K	32K	64K	64K
JPI calling convention	on	on	on	on	on	on
Multi-thread	off	off	off	off	off	on
Library ID	S	C	M	L	X	MT

Memorijski modali i flagovi za njihovo modifikovanje

TESTIRANJE PROGRAMA

VID je skraćenica od *Visual Interactive Debugger* tj. dibagar. Počev od ove verzije VID se isporučuje kao sastavni deo pakeIa ali je zato cena povećana od uvodnih 60 dolara na 200 300 VID ne unosi greške u program pa ih sam za sebe neća ni eliminisati ali omogućava razgledanje istog programa i podalaka na mnogo različitih nacina Klasične funkcije dibagera se prosto podrazumevaju ali VID može i više izvršavanja programa red po red proceduru po proceduru, do neke zaustavne tačke (breakpoint) razgledanje i neposredna izmena podalaka uz naslavljanje izvrsenja programa (interaktivno menjanje podataka) sn-manje sadržaja ekrana u datoteku radi potonje analiza razgledanje ne samo generisanog mašinskog koda nego i sadržaja registara posmatranja vrednosti promenljive lokom izvršenja programa postavljanja memorije na odredenu vrednost pre izvršenja programa, brz prelaz iz jednog dala programa u drugi snimanje slanja celog sistema Jako da se rad sa VID-om može prekinuti i kasnije nastaviti ild

Da bi se VID mogao uopšte upotrebiti potrebno je aktivirati opcije za generisanje MAP i DBD datoteka kao i za generisanje brojeva linija VID se može pokrenuti i iz editora a ne samo iz DOS-a Moguće je VID aktivirati i kao *post mortem* dibager tj da prikaže stanje memorije u trenutku kada je došlo do prekida rada

Nacin rada je isključivo preko menija a VID rezultate prikazuje u prozorima

BIBLIOTEKE MODULA

Povezivanje prevedenog programa sa sistemskim datotekama u svim *TopSpeed* jezicima rasano na najbolji moguci nacin kroz izvešlo povezivanje (*smart linking*). Umesio da se u program ubacuju sve postojeće sistemske procedure i veštim povezivanjem uzimaju se u obzir samo one procedure koje u lom programu zaista postoje Za *TopSpeed* jezike svejedno je da li se procedura nalazi u jednoj velikoj ili u 200 malih bibliotekama pa je uobicajena praksa u istu biblioteku uključivati i srodne i nesrodne funkcije.

Po ugledu na C, modula-2 je „mali" jezik, a sve sto je korisno nalazi se u spoljnjim modulima Svaki proizvodač može ponuditi sopstveni sistem modula jer iako postoje standardi za modulu 2 pomoćni moduli nisu definisani Kao uslugu pocetnicima a i radi kompatibilnosti sa literaturom u *JPI Modulu-2* ubačeni su sledeći Virtovi moduli **Terminal FileSystem**, **InOut RealInOut** i **MathLib0** Njih naravno, niko neće koristiti jer za svakodnevni rad na PC računaru nude se sasvim drugi moduli **AsmLib BiosIO CLB FIO FIOR FloatExc Graph IO Lib MATHLIB MsMouse Process ShtHeap Storage Str SYSTEM** i **Window AsmLib** grupiše potprograme kojima je jedino srodno to što su napisani direktno u asembleru Slično, **CLB** omogućava pristup potprogramima iz C-a i asemblera Ta dva modula su pseudomoduli a mnogi drugi moduli ih poziva ju

BiosIO omogućava direktan pristup IO delu BIOS-a Služi za analiziranje rada tastature i otkriva da li su pritisnuti lasteri Shift, Ctrl i Alt *Modul* **FIO** omogućava pristup datotekama pod DOS-om a **FIOR** je isto to samo za preusmerene (redirected) datoteke **FloatExc** podržava matematički koprocesor **Graph** podržava uobicajene grafičke standarde (CGA EGA VGA harkules) uz razne grafičke operacije (lačke ili nije krugovi) modul **IO** je za unos i ispis podataka na standardnim **IO** uređajima (slova brojevi tekst), modul **Lib** je skup potprograma kojima je jedino zajedničko to što nemaju niceg zajedničkog (sort-nanje generisanje slučajnih brojeva prekidi pod DOS-om analiza DOS naredba „dugački" skokovi – skokovi iz jedne procedure u drugu adresna aritmetika zvuk)

Modul **MATHLIB** sadrži uobičajene matematičke funkcije (trigonometrija logaritmi konverzije realnih brojeva u BCD formal posebne naredbe za koprocesor) **MsMouse** je interfejs za miša **Process** omogućava multijasking unutar korisnikovog programa uz sinhronizaciju procesa pomoću semafora, **ShtHeap** omogućava da korsnik sam organizuje svoju dinamičku memoriju (**heap**) pod uslovom da se zna da će za to jo biti potrebno manje od 64K. **Storage** je standardna dinamička memorija (zauzimanje i oslobadanje memorije lokom izvrsenja programa) **Str** je za rad sa lakslom (string) modul **Window** je za prozore dok je modul **SYSTEM** specifičan jer ne koristi nijedan drugi modul Osim toga, neke osobina u modulu **SYSTEM** zavise direktno od prevodioca, pa se tu zapravo radi o jos jednom pseudo modulu Inače u njemu se nalaze naredba i strukture podalaka specifične za porodicu procesora 80×86 kao i podrška za konkurentne procese

TECHKIT

Isporučeni sistem modula pokriva sve vrste uobičajenih potreba pod DOS-om Ako zelite da napišete obračun plata za svoju firmu neko klasičnu bazu podalaka Jehnički proračun bilo koje vrste ili bilo koju jednokorisnicku DOS aplikaciju *JPI Modula-2* je odličan izbor Ipak, za potpuno iskorišćenje DOS-a dobro dode *TechKit* On se sastoji iz nekoliko pomoćnih programa (WATCH TSPROF TSDA asembler) i modula (RS PMD TSR ild) TSPROF je profajler tj program koji analizira druge programe i objavljuje frekvenciju izvrsavanja pojedinih delova programa Možemo ga upotrebiti za bilo koju vrstu programa pod uslovom da posloji odgovarajuca MAP datoteka TSDA je disasembler – program koji od „gomile" mašinskih naredbi regenerise program na asembleru

WATCH je vrlo interesantan program instalira se kao prilajeni (*memory-resident*) program i odatle izvestava o rezultatima poziva DOS funkcija Nema potrebe posebno objasnjavati da se pod DOS-om sva radi upravo pozivima njegovih funkcija, a programom WATCH programer može da vidi stanja registara procesora pre i posle poziva To je naročito značajno kod pisanja prilajenih (rezidentnih) programa jer su tu mogućnosti testiranja programa najslabije

Tehnički gledano svakoj DOS funkciji odgovara određeni broj u okviru prekida 21H WATCH ne može pratiti „prijavo" pisane programa npr programe koji direktno pišu po ekranu ili desku WATCH grupiše pozive DOS-a u 15 kalegonja IO operacije za pojedinačne znake pristup disku imenici (directories) otvaranje zatvaranje čitanje, upis, pretraživanje i brisanje datoteka razne DOS usluge, datum i vreme početak i kraj programa, zauzimanje memorije mreze, i „funkcije koje ne uspevaju" Normalno ako ne znamo tačno u čemu je problem biracmo čitave grupe funkcija a kada suzimo izbor precićamo na poslednju grupu jer to su bas funkcije koje pogrešno rada

Najveću prmenu WATCH ima kod testiranja pritajenih programa Tu vrstu programa je relativno teško pisati pa *TechKit* sadrži modul TSR Korisnik treba samo da napiše proceduru koja čini telo prilajenog programa – sve ostalo će modul TSR učiniti sam To je već velika pomoć ali sve opasnosti ostaju prilajeni program će vrlo lako krahirati i sa TSR modulom ako programer ne zna tacno šta radi

TopSpeed Assembler je jednostavan i brz Nije MASM kompatibilan već su sintaksa i leksika slične samoj *TopSpeed Modulu-2* Generise se OBJ datoteke koje mogu koristiti svi *TopSpeed* jezici

Posloji i poseban modul za rad sa serijskim priključkom RS-232 To nije komunikacioni modul jer nema nikakve protokole editor i slicno Ipak demo program RDEMO je dovoljno jak da pozova zadali telefonski broj i da se uključi u na primer „Sezam"

TechKit objasnjava kako bi se sa ovom verzijom *TopSpeed Module-2* mogli pisati *Windows 3 0* programi Osim što teoretske mogucnosti za to postoje na ovu ideju treba odmah zaboraviti programiranje za *Windowse* nikakve veze nema sa programiranjem za DOS

Daleko najinteresantniji deo *TechKita* posvećen je DLL bibliotekama Čudno je da osnovni pakel podrzava DLL biblioteke ali da se kompletna dokumentacija nalaz u *TechKitu* kojeg treba dokupiti posebno

DLL BIBLIOTEKE

Tradicionalni operativni sistemi – a DOS je baš takav – očekuju da se program sastoji iz jedne datoteke Ako se koriste potprogrami iz sistemskih biblioteka sve ih valja uključiti u krajnji program i taj proces se zove povezivanje (*linking*) Ako se biblioteka promani proces povezivanja mora se ponoviti Prevedeni program u sebi nosi i deo te biblioteke Povezivanje je statičko a posledica je da se na istom disku ponavljanju delovi istih biblioteka Lek za tu situaciju je dinamičko povezivanje (*dynamic linking*) I ono se javlja u dva oblika kao kasno povezavanja (*late binding*) i kao zajednicki pristup naredbama i podacima (*shared code* and *data*). Kasno povezivanje znači da se prava verzija potprograma ne zna sve do samog izvrsenja tog potprograma Zajednicki pristup memoriji znači da nema potrebe da se u memoriji ikada nade više od jedne verzije nekog potprograma U oba slučaja umesio samog potprograma u memoriji se nalazi pokazivac na potprogram a sam potprogram ce biti u posebnoj biblioteci prikladno nazvanoj DLL – *Dynamic Link Library*

Naravno, DLL biblioteke ne treba koristiti svugde ali je činjenica da ovaj pristup kreranju programa postaje dominantan naročito pod *Windowsima* i pod OS/2

Programera što se tiča nikakvih razlika nema u pisanju programa a jedina razlika u prevođenju je što umesto DOS EXE treba staviti DynaDLL u jednoj od opcija editora

OBJEKTNO PROGRAMIRANJE

U jezičkom smislu najveća novina u ovoj verziji je nekoliko novih rezervisanih reči za objektno programiranje Zastupljeni su osnovni pojmovi klasa objekat metodi nasleđivanje virtualni metodi i slicno Nove službene reči su CLASS, SELF i VIRTUAL Virtualni metodi su slični onima u *Turbo Pascal*-u posloji posebna tabela kroz koju se poziv ispravnog metoda preslikava u izvršenje potprograma Zalo je pridodata nova standardna funkcija MTA koja vraća pokazivač na labelu virtualnih metoda

Nastavlja se na strani 69

FORTRAN 90: USVOJENO KOMISIJSKI

Poslednje raapoložive informacije govore da će *Fortran 90* biti među nama vrlo skoro. Međunarodna zajednica je već usvojila novi atandard. Očekuje se aamo da ga priznaju i SAD, koje pak predlažu da *Fortran 90* na zameni atari *Fortran 77*, već da ide uz njega kao dodatni atandard. Ovo bi moglo umanjiti efekte usvajanja novog standarda, ali kompromisi u fortranu ionako nisu ništa novo. Ovaj jezik „za prava matamaričare" su oduvek krojile – komisije.

Podsetimo se ukratko, novije istorije fortran standarda (preskočićemo *Fortran IV* i *Fortran 66*) *Fortran* standardi su uvek stvarani uz velike poteškoće, all je ovaj poslednji pokazao kako nazvoj uz pomoć komitete (reč je o X3J3 komitetu) može biti izuzetno težak Izgleda da je oblikovanje programskih jezika jedno od područja gde demokratija nije baš najefikasnija.

KRATKO PODSEĆANJE

Fortran 77 ima svojih mana (i to priličnih), neki elementi ovog standarda usvojeni su u poslednjem trenutku, a sam jezik je postao zastareo još i pre nego što je standard i usvojen Mnogi veruju da je saglasnosl oko *Fortran 77* standarda postignuta samo zato da se stvon pnlika da se posao obavi ponovo, eli ovaj put temeljnije Tako su narednih godina rodeni *Fortran 82* *Fortran 85*, *Fortran 88* ukratko – *Fortran 8x*. Ništa kod *Fortrena 8x* nije bilo melo izmene su bile tako velika da bi bilo potrebno pola dana da se one izlože!

Ali stvari nisu kretale na bolje i već se šeputalo da je ono „x" heksadecimalno *Fortran 88* je (skoro) podskup *Fortrena 77* koji je podskup Fortrane 8x, i tako redom ka sve većem brojevima Ideja o fortranu kao malom kompaktnom jeziku podržanom modulima za posebne namene odavno je napuštene Jezik je jednostavno postajao prevelik Tada su tvorci standarda rešili da ga malo „potkrešu" što je dovelo do nedoslednosti i do gubitka dragocene funkcionalnosti Srećom nije se sve na tome završilo i od tada je uneseno mnogo izmena karaktenstike su dodavane menjane i izbacivane

ŠTA JE NOVO U 90-tim?

Kao rezultat svih ovih promena, pred nama će se naći *Fortran 90*, koji, osim što u potpunosti obuhvala *Fortran 77* ima mnoštvo novih, vrlo upotrebljivih osobina. Najvažnije nove karaktenstika *Fortrena 90* su sledeće

- novi oblik izvornih (source) datoteka izvorni kod u slobodnom formatu sa linijama dugačkim do 132 karaktera;
- ! kao oznaka komantara
- & na kraju linije koja se nastavlje u sledeem redu,
- imena promenljivih do 31 znaka (uključujući i _),
- razmaci su bitni
- novi karakteri: $! „ % & < > ?
- velika i mala slova se tretiraju podjednako, osim kod znakovnog niza,
- nove **Type** deklaracije Ugradene „istražne" funkcije za posebne tipove podataka,
- **Izvedeni tipovi** – definiše lh sam korisnik
- konsnički definisano rukovanje **izvedenim tipovima**
- pnpadajući (generički) interfejsi korisnički definisanim tipovima
- Pokazivači (pointen) i **ciljevi** (*targets*) za privremeno povezivanje sa različitim objektima,
- Nove kontrolne strukture CASE struktura za izbor
- DO ENDDO,
- CYCLE i EXIT komande za kontrolu petlje
- Imenovanje kontrolnih struktura,
- Moduli za definisanje i inicijalizaciju promenljivih i nizova koje dele različiti potprogrami; za definisanje interfejs struktura, za definisanje **izvedenih tipova**;
- PUBLIC i PRIVATE entiteti;
- dinamičko alocirenje memonje veličine nizova u zavisnosti od ulaznih podaleke,
- nizovi prelpostavljenog oblika;
- automatski objekti
- operacije sa vektorima i matricama ,
- nove ugradene funkcije ,
- potprogrami interni (unutrašnji) potprogremi koji dele promenljive sa svojim „roditeljskim" potprogremom, rekurzivni potprogrami,
- označavanje argumenate potprograma kao INPUT ili OUTPUT,
- OPTIONAL argumenti,
- nove U/I kontrole

Pogledejmo malo detaljnije neke od ovih novih karakteristika

IZVEDENI TIPOVI PODATAKA

Fortran je uvek bio dobro opremljen jezik što se tiće različitih tipove podataka i tradicionalno ima pet tipove podataka poznatih kao *intrinsic* (unutrašnji ugradeni) To su INTEGER, REAL COMPLEX, LOGICAL i CHARACTER U *Fortranu 90* programeru će biti omogućeno da definiše svoje sopstvene tipove podataka. Ovakvi izvedeni tipovi podataka mogu se posmalrati kao strukture podataka pošto mogu sadržati promenljive različitih tipova Na primer, pretpostavimo da nam treba program koji će obradivati proste statističke podetke o raznim časopisima ze period od jedne godine Programu će biti potrebno da zna imena časopise broj zaposlenih mesečne traže i mesečne cene (pretpostavka je da časopisi izlaze jednom mesečno) Ove informacije se mogu čuvati u pogodno definisanoj struktun nazvanoj recimo PODACI, koja se može definisati na sledeći način

```
TYPE PODACI
    CHARACTER (LEN=20)   NAZIV
    INTEGER    BR_ZAPOSLENIH
    INTEGER,DIMENSION(12) &
    =TIRAZ
    REAL DIMENSION(12)   CENA
END TYPE PODACI
```

Uočimo „&" karakter koji nastavlje liniju u sledeem redu, dok je linija CHARACTER (LEN=20) NAZIV mogla biti napisana i kao CHARACTER*20 NAZIV ali je novi oblik opšliji

Sada, kada smo definisali tip PODACI, možemo deklarisati promenljive tog tipa Da bismo definisali promenljivu CASOPIS kao niz od 20 elemenata tipa PODACI, u program bi trebalo staviti

```
TYPE PODACI DtMENSION(20) &
  CASOPIS
```

Nad ovakvim konsnički definisanim tipovima podataka moguće je sada definisati različite operacije, u šta u ovom tekstu ne bismo ulazili.

Izvedeni tipovi podataka i njihovi interfejs operaton mogu biti definisani na vrhu potprograma u kome se konste, eli je mnogo upotrebljivije da se ti tipovi deklarišu u novoj potprogramskoj strukturi nazvanoj MODULE. Različiti moduli mogu se konstruisati za različite familje tipova podataka, a pojedinim modulima može se pristupiti unutar izabranih potprograme pomoću USE naredbe MODULE struktura može se iskoristiti takode i za definisanje i inicijalizaciju skalamih i niz-promenljivih, i preporučljivo je njeno konščenje umesto COMMON blokova i BLOCK DATA naredbi

POKAZIVAČI I CILJEVI

Fortran 90 uvodi pokazivačke promenljiva u fortran jezik. Promenljiva koja je definisana keo POINTER može se upotrebljavati na isti način kao i neka obična promenljiva, all takode može biti iskorišćena i drugačije Pokazivač je promenljiva koja može biti privremeno (dinamički) povezana sa nekim objektom podataka (date object) Objekt podetaka mora biti istog tipe i istih dimenzija kao i pokazivač Pokazivač uvek ima jedno od tri stanje On može biti povezen sa objektom podataka, čak i ako taj objekat nije trenutno definisan Može biti bez vrednosti (null), što znači da trenutno ne ukazuje ni na šta Ili može biti nedefinisan, što je uvek slučej kada je pokazivač prvi deklarisan Smatra se lošom programerskom praksom da se pokazivači ostavljaju nedefinisanim. Pomoću NULLIFY neredbe možemo nedefinisane pokazivače pretvoriti u „null" pokazivače Da bi pokazivač mogao da pokazuje na neki objekt podataka, taj objekt podataka mora bih deklarisan kao TARGET (cilj)

Kao primer, pogledajmo sledeći kod koji ne radi ništa pametno osim što baca malo svetla na upotrebu pokazivača i cilјeva

```
REAL POINTER  P1,P2
REAL,TARGET  V,R
R = 9.25
V = R
P1 => V          !povezivanje
P2 => P1         !povezivanje
R = 6.0
P2 = R           !obična dodela vrednosti
```

Koje vrednosti imaju P1, P2, V i R posle izvršenja ovog programa? Odgovor je da svi imaju istu vrednost 6.0 U petoj liniji je P1 po-

stao zamene (drugo ja, alias) za V. Zatim je P2 poslao zamena za P1, a samim tim i za V. Zato kada P2 uzme isti vrednost kao i R, P1 i V takođe uzimaju vrednost R, a pošto je R=6 0 u svim promenljivima je ta ista vrednost. (Ko je ostao živ može ovo još jednom da pročita da utvrdi gradivo.) Znak uzvika upotrebljen u gornjem primeru označava početak komentara. Pokazivači su višestruko korisni. Na primer, rutina koja sortira podatke obično će biti mnogo brža ako se sortiraju pokazivači na podatke, nego sami podaci. Direktno sortiranje podataka prouzrokuje njihovo premeštanje po memoriji, dok sortiranje pokazivača jedino dovodi do izmena veza.

Najvažnije osobina pokazivača je ta da se pomoću njih može dinamički alocirati i dealocirati memorija pomoću ALLOCATE i DEALLOCATE naredbi. To znači da se za jedan podatak može kreirati prostor u memoriji gde će on biti smešten, pri čemu će na njega ukazivati jedino pokazivač iz odgovarajuće ALLOCATE naredbe. Taj podatak može biti i izvedenog tipa. Naredba DEALLOCATE oslobađa zauzelu memoriju kada podatak više nije potreban. Pre *Fortrana 90* jedino je preko niza (jednodimenzionog ili višedimenzionog) bilo moguće sakupiti na jednom mestu podatke povezane po nekom osnovu, ali su i tada podaci morali biti istog tipa. Uvođenjem izvedenih tipova podataka omogućeni su nizovi struktura podataka, sa komponentama različitih tipova. Međutim, kruta struktura niza nije uvek poželjna. Ako je potrebno umetnuti podatak na određenu lokaciju u nizu koja je već zauzeta, tada se moraju svi podaci na lokacijama višim od te pomeriti da bi se napravilo mesto. A ako je niz popunjen, tada nema načina da se doda novi podatak. Dinamičko alociranje pomoću pokazivača omogućava grupisanje podataka na nov način u obliku vezanih lista (*linked lists*), slabala i drugih. Svaki član vezane liste sadrži osim podatka i pokazivač na sledeću člana liste, a po potrebi i pokazivač na prethodnog. Da bi se ovde umetnuo novi podatak, potrebno je samo izmeniti pokazivač člana koji je „ispod" mesta umetanja da ukazuje na novog člana i da se pokazivač novog člana usmeri na člana „iznad" mesta umetanja.

PROGRAMSKE PETLJE

Kontrolne struktura u *Fortranu 90* su pojačane i proširene. Jedan od problema u složenijim polprogramima je stvaranje mnoštva numeričkih labela koje su potrebne za DO petlje i GOTO naredbe. Održavati logičan redosled brojeva može biti zamoran posao i često se zanemaruje zbog čega je kod bude naporan za praćenje i težak za održavanje. IF... THEN... ELSE... ENDIF struktura jeste jedna logične sekvence koda, ali je teško upariti IF... ELSE sa odgovarajućim ENDIF-om kada su IF-ovi „ugnežđeni" čak i kada je kod propisno razuđen. *Fortran 90* za ovakve situacije nudi više rešenja. DO... ENDDO petlja uklanja potrebu za numeričkim labelama, a poseduje i CYCLE i EXIT naredbe za prenos kontrole na ENDDO naredbu i za izlaz iz petlje, respektivno. U cilju lakšeg praćenja komponenti kontrolnih struktura programer im može pridružiti opisne labele tj. imena, pa i to doprinosi čitljivosti koda. Opisne labele su opcione i nisu neophodne u slučajevima kada se kontrolna struktura proteže na samo nekoliko linija. Na primer

```
DO 20 I=1, N
IF (I .EQ. 5) GOTO 20
...
20 CONTINUE
```

može biti zamenjena sa

```
DO I=1, N
IF (I .EQ. 5) CYCLE
...
ENDDO
```

Opisnu labelu treba upotrebiti ako se DO petlja proteže na veći broj linija, ili ako je ugnežđena

```
petlja      DO I=1, N
              ...
              ...
            ENDDO petlja
```

Jedna imenovane IF... THEN... ELSE... ENDIF struktura mogla bi izgledati ovako

```
proces: IF (opc.EQ.1 .OR.opc.EQ.3') THEN
          ...
        ELSE IF (opc.EQ.2')
               THEN proces
          ...
        ELSE proces
          ...
        ENDIF proces
```

Ovde bi ovde nova CASE konstrukcija mnogo bolje „pasovala":

```
proces: SELECT CASE(opc)
          CASE ('1' '3')    proces
          ...
          CASE ('2')        proces
          ...
          CASE DEFAULT proces
        END SELECT   proces
```

CASE konstrukcija može se upotrebljavati za izbor između vrednosti izraza tipa CHARACTER, LOGICAL ili INTEGER. Ako je prva linija CASE konstrukcije imenovana, onda i zadnja (END SELECT) linija mora imati isto ime, dok imenovanje linije između ove dve nije obavezno.

A GDJE JE SIR ?!

U *Fortranu 90* ima još mnogo toga. Samo sira tali. Na primer, tu je naredba INCLUDE koja se dosad nalazila kao proširenje kod mnogih *Fortran 77* kompajlera. Ona je korisna utoliko što omogućava da se COMMON blok napiše samo jednom u svojoj sopstvenoj datoteci, a da se zatim uključuje (*include*) u sve polprograme koji ga koriste. Oko uvrštavanja ove naredbe u standard bilo je mnogo rasprave, jer su je druge nove karakteristike *Fortrana 90* učinile suvišnom. Na kraju je INCLUDE ipak postao deo novog standarda.

Fortran (stari) je imao neke ozbiljne nedostatke, što je i logično jer se inače ne bi toliko trudili da ga poboljšaju. Od svih najviše frustrira statičko alociranje memorije tj. da se veličine svih nizova moraju biti eksplicitno postavljene bez obzira da li se radi o nizovima u potprogramima ili glavnom programu. Lek za gore navedeno je dinamičko alociranje memorije koje je u *Fortranu 90* ostvareno putem definisanja alocirajućih nizova. Kada veličine tih nizova postanu poznate ze odgovarajući skup podataka one (veličine) se dodeljuju nizovima preko ALLOCATE naredbe

```
REAL A,B,C
ALLOCATABLE A( ),B( , ), &
C( , )
! dimenzije moraju biti zadate
READ(5,*) L,M,N
ALLOCATE (A(L), B(M,N), &
C(N,L))
```

Primetimo da se u ALLOCATABLE naredbi deklarišu samo dimenzije, a ne i veličine nizova, što je dosad bilo puste želje fortran programera. Uz pomoć ovih novih naredbi više neće biti neophodno da se pri prenosu niza u potprogram u listi argumenata šalje i veličina niza, a takođe će i lokalni nizovi u potprogramima moći biti promenljivih dužina. To će pisanje potprograma učiniti mnogo fleksibilnijim, a smanjiće i veličinu liste argumenata što će sve olakšati održavanje koda. Kada alocirajući niz nije više potreban, memoriju oslobađamo sa DEALLOCATE. Još jedan način da se smanji veličina liste sa argumentima je da se neki od argumenata deklarišu kao OPTIONAL (opcioni). Ovi OPTIONAL argumenti mogu tada biti isključeni iz liste argumenata kada nisu potrebni. Meda je ovo dobrodošla mogućnost. OPTIONAL argumenti moraće se upotrebljavati vrlo pažljivo, da bi se izbegla neslaganja argumenata u pozivnoj listi i listi pozvanog potprograma.

Još jedna dugo očekivana novina koju donosi *Fortran 90* je uvođenje vektorskih i matričnih operacija. One bi trebalo da poboljšaju optimizaciju koda, a takođe će smanjiti i veličinu i složenost izvornog koda. Na primer, ako su A, B i C realni (REAL) nizovi sa istim dimenzijama, a veličinama onda je A=B+5.0 + C/10.0 + 6.0 ispravan izraz. Prvi elemenal od A će biti jednak prvom elementu od B pomnoženom sa 5, plus prvom elementu od C podeljenom sa 6 plus 6 i tako dalje za sve elemenle nizove A, B i C.

Sve ugrađene funkcije *Fortrana 90* (pored onih nasleđenih iz *Fortrana 77*) imaju generičke imena, a gde je to pogodno, openšu i sa skalarnim i sa niz-promenljivima. Jedan broj novih ugrađenih funkcija uveden je posebno za operacije nad nizovima, skalarni proizvod vektora, transponovanje matrice i množenje matrica. Osim ovih, tu su i mnoge druge nove ugrađene funkcije koje će omogućiti konstrukciju i manipulaciju sa nizovima, zatim konstrukciju procedura za rad na nivou bitova, generatora slučajnih brojeva i pristup časovniku realnog vremena.

Literatura

Pri pisanju ovog teksta korišćeni su sledeći članci:

Getting ready for fortranave 90 (Program NOW, Decembar '90) Autor profesor Tim Ward, koji profesionalno radi na razvoju fortran-baziranih sistema, a takođe je i autor knjige *Fortran and the Art of PC Programming*.

Derived data-types in Fortran, (Program NOW, Januar '91) Autor profesor Tim Ward.

Getting to grips with Fortran 90 (Program NOW, Februar '91) Autor profesor Tim Ward.

Fortran 90 - you can get there from here! (Program NOW, Jun '91) Autor: Peter Anderton, direktor za marketing u Numerical Algorithms Group Ltd. koja lansire Fortran 90 kompajler u junu '91.

I NA KRAJU...

Fortran 90 ima nove, moćne osobine koje su do sada pripadale jezicima kao što su C, C++ i ADA. Ove nove karakteristike će olakšati programiranje u fortranu, a prava korist će biti u efikasnijem razultujućem mašinskom kodu. Ipak, ni jedna od ovih novina nece dramatično izmeniti način na koji se pišu fortran programi. Mogli bismo pomisliti da je to i dobro, ali, na žalost, modularni pristup programiranju koji je uveden baš u fortranu nije više dovoljan da zadovolji zahteve za sve većom funkcionalnošću naših programa.

Nastavlja se na stranu 77

KRUNA ZA KRALJICU NAUKA

Ako se na bilo koji način bavite matematikom – kao učenik, kao profesor matematike ili fizike ili kao naučnik – i u svom radu koristite računar, softverski paket *Mathematica* je jedna od najboljih alatki koju možete imati. Pokušaćemo, kroz tekst u ovom i sledećem broju, da odgovorimo zašto se *Mathematica* smatra jednim od dva najbolja softverska paketa iz ove oblasti.

Milan Bojanović

Nije uobičajeno da se prikom predstavljanja softverskih pakela pominju njihovi auton barem iz dve razloga. Prvi, svi ioie ozbiljnji paketi delo su velikog broja autora čije bi predstavljanje odnelo više prostora nego opis samog softvera. dok se drugi liče same suštine pisanja – koliko je sva to zanimljivo potencijalnom čitaocu. Međutim, auton softverskog paketa *Mathematica* zaslužuju da se odslupi od ovog pravila: osobito njegov glavni tvorec Stephen Wolfram. Skoro da neverovatno zvuče (svakakao za jugoslovenske prilike) izvodi iz njegove biografije, navedene u uvodu knjige MATHEMATICA roden 1959. u Londonu, tehnijsku fiziku doktorirao sa 20 godina na Caltech-u, a na Univerzitetu u Illinoisu je profesor fizike, matematike i računarske tehnologije. Usput, ne ovom Univerzitetu je i direktor centra ze istraživanje složenih sistema. Rezultati njegovih istraživanja u matematici i fizici primenjuju se u biologiji, mahanici fluida... Savetnik je u nizu vodećih firmi za proizvodnju računara (Bell Labs' Thinking Machines Corporation ...) Slične biografije imaju i ostalih sedam autora što na svoj način vrlo ubedljivo govori o samom proizvodu.

Šta je zapravo Mathematica? Na prethodno pitanje vrlo je teško dati kratak odgovor a ostali precizan. Sam naziv softverskog pakela izgleda prilenćiozno, ali nakon duzag rada i komparacija sa ostalim pakelime iz ove oblasti došli smo do nepokolebljivog zaključka – reč je o vrlo moćnom alatu u matematici koji konsniku daje rešenja u najrazličitijim oblicima. U ovom članku biće reči o poslednjoj verziji ovog softvera, koji se prodaja pod oficijalnim imenom *MS-DOS 386 Mathematica V 1.2*.

NA PRVI POGLED...

Bez obzira na našu neviknutost na vrhunski pisani uputstva ze rad sa softverom, kakva samo mogu pripremiti Hewlett Packerd, DEC, IBM ili Tektronix, uputstvo ze rad sa *Mathematica*-om nam je poiipuno zbunilo. Prva pomisao je bila da se radi o nekom dopunskom udžbeniku uz softver, koji se uzgred isporučuje u lepo zapakovanoj kutiji prekrivenoj masom cifara (više o njima kasnije). Međutim, uputstvo za rad je profesionalno oblikovene knjiga u vrlo kvalitetnoj štampi koju je izdala poznata knjižarska kuće Addison-Wesley. Inače, ova knjiga se može bar po tvrdnji iz nje kupiti u odgovarajućim knjižarama. Drugi dao paketa *Mathematica* je „podpakel" se knjigama *User's Guide, Reference Guide, Updates, Release Notes, Technical Report* knjižica se spiskom dopunskih funkcija koje su razvili razni auton i koje se mogu pozvati iz glevnog programa.

Naravno, tu je i neizbežan registracioni karton, kao i nesvakidašnja narudžbenica za majicu sa grafickim aplikacijama uradenim u *Mathematica*-i (košta cca 12 USD $). Napokon, posle kraćeg „kopanje" po papirima, dolazi se i do prostora sa dva puta po dva diskete (od 3.5"i 5 1/4") na kojima se nalazi distribucija softvera. Da bi se instalirao softverski paket *Mathematica*, neophodno je obezbediti sledeću PC platformu: 80386 procesor matematicki koprocesor zavisno od kupljana verzije (bez koprocesora Intel i kompatibilni koprocesori ili Weitek 1167 i 3167), MS DOS, 640 + 1024 K RAM-a i najmanje 6 MB slobodnog prostora na hard disku, pri čemu treba voditi racuna da se ne podrzava tzv. Lotus-Intel-Microsoft standard za rad sa *expanded* memorijom.

Procedura instalacija je slična kao kod ostalih softverskih pakela ovog „kalibra" stralja sa instalacioni program MINSTALL koji vrši dekompresiju programa i prenosi ih u prethodno kreirane podrekorijume PACKAGES i FONTS. U prvom od njih nalazi se se niz korisnih programa koji se direktno pozivaju iz *Mathematica*-e dok su u drugom fontovi za grafičke prikaze. Tokom instalaciona procedura u glavnom direktorijumu osim izvršnih programa formiraju se i tri komandna programa: MATH.BAT – glavni komandni program kojim se pozivaju izvršni programi HARDCOPY.BAT – komandni program koji šalje *PostScript* datoleke na printer, i DISPLAY.BAT – program koji salje *PostScript* datoleke na ekran.

Uz klasičnu poruku da se promene na instalacji mogu izvršiti ponovnim startovanjam instalacionog programa za početak rada dovoljno je otkucati komandu MATH. Na ekranu se posle podueže čekanje pojavljuju imena autora softvera i vrlo kratka poruka

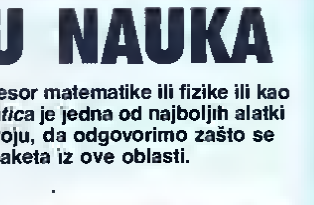
```
In  [1] =,
```

jer korisnici MS DOS operativnog sistema nemaju pnvilagiju rada sa prozorima, kao vlasnici mekintoša ili SUN radnih stanica.

Kako je sve moguće koristiti ovaj softver?

...SVE NAJBOLJE

Prvo *Mathematica* se može koristiti kao klasičan elektronski kalkulator za standardne aritmetičke operacije jednostavno unesete u program 15 + 10 i *Mathematica* vam odgovara da je zbir 25. Svakako da nikoi iole razuman, nje spreman da za ovakvu kategoriju problema neveslka u softver cce 600 USD $. Međutim i za „kalkulator" probleme *Mathematica* daje neverovatna rezultate u smislu preciznosti. Na primer, kvadratni koren broja 11 dobijen na kalkuletoru sa preciznošću od osam decimela iznosi 3.31662479. Kvadratni koren istog broja može se zahtevati od *Mathematica*-e i sa preciznošću od 20, 100, 1000 do neverovatnih (verujte mi na reč) 5000 decimalnih mesta. Za svega nekoliko sekundi (primeri su rađeni na računaru ACER 1200 sa 486 procesorom i 4MB RAM-a) ekran je bio preplavljen ciframa iza decimalne tačke uz popuku da je sva OK! Na slici 1 je prikazano kako izgleda kvadratni koren broja 11 sa prvih hijadu cifara iza decimalne tačke.



Slika 1 Kvadratni koren broja 11 sa prvih hijadu cifara iza decimalne tačke

Slične zadovoljstva vam priređuju rezultati tipa

$$3^{30} = 717897987691852588770249$$

koji kod klasičnog kalkulatora imaju skraćenu notaciju, na primer

$$3^{30} = 7.17897987 * 10^{23}$$

Na zahtev da se odredi koren iz broja –9 *Mathematica* vam nece poslati poruku o greški već ce dati rezultat 3 i, gde i predstavlja imaginarnu jedinicu. Zastupljene su i ostale operacija sa kompleksnim brojevima.

Uz pretpostavku da je ovaj softver kupljen ponajmanje za opisana numeričke operacije necemo njima posvećivati veću pažnju. Napomenimo još samo da je moguće sa *Mathematica*-om rešavati probleme iz domene matričnog i integralnog računa, Furieove transformacije, numeričke minimizacije, specijalne funkcije, nalaženja korena polinoma, sistema jednačina i slično.

Mathematica ima još niz korisnih funkcija. Na primar, moguce je prikazati funkcije u 2-D i 3-D postoru (slika 2) uz prikaz svih relevantnih podataka na grafici ma. Obezbeđene su funkcije sencenja površi, uklanjanja nevidljivih linija, pogleda sa različitih strana... Takođe, ako je potrebno vizualizirati matematičke relacije koje se menjaju lokom vremena, konsniku je na raspolaganju opcija za animaciju. Uspešna animacija u visokoj razolucij zahteva odgovarajuću grafičku kartiou (za „double buffering") i velike memorijske resurse personalnog računara.

Mathematica je istovremeno i programski jezik interpreter, a keo posebna poslastica za programere u C ili fortranu, nudi se generisanje odgovarajućih programskih kodova za matematičke relacija napisane korišćenjem jednostavne sintakse *Mathematica*-e. Takođe, vrlo lako (sa svega jednom komandom) se pripremi i format za obradu u TEX-u.

A sad nešto više o mogućnostima *Mathematica* koje ce sigurno obradovati svakog studenta (varovatno ne samo njih) prirodno-matematičkih i tehničkih nauka.

VIŠE OD KALKULATORA

Pošto je iz prethodnih nekoliko primera sasvim jasno da *Mathematica* daje izuzetne rezultate u domenu numeričkih operacija, poka-

zacemo da je njena najveća vrednost ipak u nečem drugom – operacijama iz oblasti simboličke matematike. Ovo znači da *Mathematica* može opensati kako sa brojevima tako i sa formulama. Evo jednostavnog primera sa formulama

```
In  [1] = 4x + 5x - 3 + 1
Out [1] = 9x - 2
```

Iz navedenog primera vidimo da **x** nema određenu numeričku vrednost a ipak je izvršeno sabiranje, odnosno pojednostavljivanje navedenog izraza. Da ne bismo odmah preplašili čitaoce Računara nerezumljivim jednačinama prikazaćemo i rešiti više ilustrativnih primera koji na uverljiv način govore o mogucnostima softverskog pakata *Mathematica*.

Posmatrajmo, na primer, padobranca koji je iskočio iz aviona (slika 3). Da bi dobili sve relevantne parametre njegovog pada na „majku" Zemlju, prvo je neophodno napraviti odgovarajuci matematički model. Taj model podrazumeva definisanje jednačine koja na veran način opisuje njegovo kretanje od momenta iskakanja do pada na zemlju. Rezultujuću silu **F** koja deluje na padobranca, dobijamo kao razliku sila Zemljine teze **Fd** (**Fd**=**m**•**g** – **m** je masa padobranca a **g** gravitaciona konstanta) i sile otpora **Fu** (**Fu** = –**c**•**v** – **c** je koeficijent otpora koji je linearno proporcionalan brzini pada **v**)

F = **Fd** + **Fu**

Ako se prisetimo II Njutnovog zakona (da na kažemo Njutnove jabuke) možemo dobiti formulu koja u svakom trenuktu vremena **t** daje promenu brzine padobranca shodno silama koje deluju na njega

$$\frac{dv}{dt} = g - \frac{c}{m} * v$$

Egzaktno rešenje ove diferencijalne jednačine, o kojima će biti reči kasnije je

$$v(t) = \frac{gm}{c} * \left(1 - e^{-\frac{c}{m} * t}\right)$$

Svakako da spada u nizi srednjoškolski nivo znanja matematike računanje brzine padobranca iz date formule ako zadamo masu, gravitacionu konstantu i koalicijent otpora vazduha. Međutim problem se komplikuje ako želimo da odredimo koeficijent otpora za zadatu brzinu i masu padobranca (podrazumeva se da je **g**=**9.81m/s** = **const**). Na postoji put da koeficijent **c** izolujemo sa jedne strane znaka jednakosti (kao što smo to uradili sa **v**) i izračunamo njegovu vrednost. Dakle, imamo izv problem nalaženja korena implicitne jednačine

$$f(c) = \frac{gm}{c} * \left(1 - e^{-\frac{c}{m} * t}\right) - v$$

Ovaj problem se u numeričkoj analizi rešava grafičkom bisekcionom, Njutn Relsonovom (Newton Raphson) i sličnim metodama. U *Mathematica*-i je dovoljno uneti sledeci komandu

```
In  [1] =[FindRoot  g*m/c(1-Exp[-c*t/m])-v
== 0, {c  1000 }]
```

Uz prelposlavku da smo uneli masu padobranca od 66.1 kg, da pada brzinom od 4 m/s i da je u vazduhu 7 sekundi dobijamo da je rezultat rada prethodna komande koeficijent otpora c = 11.6431 kg/s (usput, ako imale problema sa konvarzijom jedinica, *Mathematica* vam omogućuje da definišete jedinice i na primer sabirate 3kg + 5kg). Ovim primerom smo pokazali jedan od mogućih načina za nalaženje korena polinoma. Međutim, *Mathematica* može naći koren polinoma $x^2 + ax + 2 = 0$ iako koeficijenti uz nepoznate nisu numerička vrednosti

```
In [1]  =Solve[ x^2 + ax + 2 == 0 , x ]

                                    -a + Sqrt [-8+a^2]
Out [1] := x -> ------------------
                          2

                                    -a - Sqrt [-8+a^2]
          x-> ------------------
                          2
```

Pre nado što se krene u nalazenje korena polinoma obično se prvo po nekom kriterijumu sređuju izrazi. Ovim softverom se mogu na najrazličitije načine kreirati formule shodno stavu da je pisanje formula jedna vrsta „umetničke" delatnosti u matematici. *Mathematica* nam sledeci polinom može dati u vise oblika zavisno od matematičkog dela ne kažemo umetničkog talenta konsnika ovog pakata

```
In  [1]  = y = (x-4)(x-3)(x-2)(x-1)
Out [1]  = (-4 + x)(-3 + x)(-2 + x)(-1 + x)
```

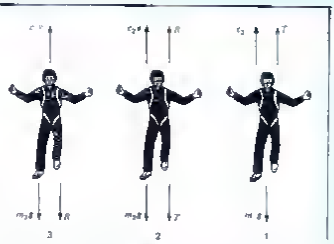
Pošto nije uobičajano u ovoj formi prestavljati funkcije **y**=**f(x)** zatražimo od matematika (komandom **Expand y**) da izmnoži sve članove prethodnog izraza. Rezultat ja prikaz funkcije **y** u kondenzovanoj formi

```
In  [1]  = Expand [y]
Out [1]  = 24 - 50x + 35x^2 - 10x^3 +x^4
```

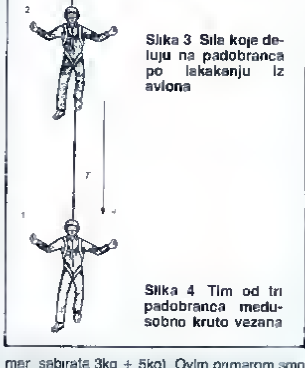
Na ovaj način se može izvršiti množenje ili deljenje polinoma i još niz operacija iz simboličke matematike a posloji više kriterijuma da se grupišu ili izdvoje pojedini članovi u izrazu



Slika 3 Sile koje deluju na padobranca po lakakanju iz aviona



Slika 5 Sile interakcije na tim padobranaca



Slika 4 Tim od tri padobranca međusobno kruto vezana

Na primer, moguće je izdvojiti iz polinoma listu koalicijenata, broj članova polinoma, najveci stepen i sl. Ove operacija su pogotovo korisna kada se radi sa racionalnim funkcijama

Zakomplikujmo život našem padobrancu od 70 kg (malo se poravio radi lakseg racuna!) i koeficijentom otpora 10 kg/s. Neka je neposredno posle njega skočio padobranac mase 60 kg, sa koeficijentom otpora 14kg/s, a potom drugi padobranac mase 40 kg sa koeficijentom otpora 17 kg/s. Pretpostavimo da je ova ekipa padobranaca povezana krutim štapom bez mase (radi lakšeg računanja) i da padaju brzinom od 5 m/s (slika 4). Neophodno je odrediti sile u svakom kvazi-štapu i ubrzanje kojim se približavaju Zemlji. Na slici 5 prikazana su sile koje deluju na padobrance. Dakle, sa slike se vidi da treba odrediti sile **T** i **R** i ubrzanje **a**. Postavimo sistem od tri jednačine sa tri nepoznala kojim se opisuje ovaj problem

$m_1g - T - c_1v = m_1a$
$m_2g + T - c_2 - R = m_2a$
$m_3g + T - c_3 + R = m_3a$

Na osnovu zadatih vrednosti matrična forma **M** • **X** = **b** ovog sistema ima sledeći oblik

$$\begin{vmatrix} 70 & 1 & 0 \\ 60 & -1 & 1 \\ 40 & 0 & -1 \end{vmatrix} \begin{vmatrix} a \\ T \\ R \end{vmatrix} = \begin{vmatrix} 836 \\ 518 \\ 307 \end{vmatrix}$$

Sa sledeće In komande *Mathematica* vam daje rešenje da je ubrzanje a=8.594 m/s², T = 36.76 N, a R = 34.41 N

```
In  [1] = M = {{70  1  0}, {60, -1  1}  {40
0  -1}}
In  [2] = M {a, T, R} == 636, 518, 307
In  [3] = Solve[% {a, T  R}]
```

Naravno ovako jednostavni primari ne govore na pravi način o (ne)moci nekog softverskog paketa. Glavni ispiti na primer za sistema linearnih jednačina su u rešavanju sistema od hiljadu i više jednačina sa hiljadu i više nepoznatih, koji se obično javljaju pri strukturalnim ili termičkim proračunim konstrukcija, a što *Mathematica* se lakocom rešava. Inače, *Mathematica*-om se vrlo jednostavno nalazi determinanta matrice, vrši njeno transponovanje ili inverzija. Takoda, obezbeđeno je množenje matrica matricom vektorom ili skalarom

OPRAVDANJE POHVALA

Vretmo se sada domenu vise matematike, i argumentujmo uvodne pohvale. Skoro da ne-

Nastavlja se na strani 39

LOTUSOVE ALATKE

Za razliku od avojih konkurenata, koji to rade u okviru osnovnih paketa, Lotua već duže vreme snagu svojih proizvoda uvećava, na naše zadovoljstvo, kroz takozvane Add-In programe. To su dodatni, posebni, programi koje prave nezavisni proizvođači, a savršeno se slažu sa originalnim programom i omogućuju korisnicima da zadovolje svoje potrebe.

Vladimir Stamenović

Neki od tih dodatnih nezavisnih programa su zahvaljujući svom kvalitetu postali sastavni delovi najnovijih verzija Lotusovih tabela. Ta kav je slučaj sa programima kao što su *Always Wysiwyg* i *Viewer*. Danas na svetskom tržištu ima na hiljade Add-In programa najrazličitijih namena pomoću kojih su rešivi problemi iz raznih oblasti, a bez potrebe da korisnik sam gubi vrama na njihovo kreiranje. Ta činjenica delimično objašnjava ogromnu popularnost Lotusovih proizvoda koja, verujem, skoro neće biti ugrožena.

SOFTVERSKI DODACI

Ovakva strategija je započeta u Lotusu odavno, a 1987. godine se pojavio *Developer Tools* pomoću koga su pravljeni dodatni programi za 1-2-3 verzija 2.x i *Symphony*. Namenjen je bio kompanijama koje prave dodatne programe za prodaju. Mada ovi alati nisu bili ni malo prijateljski nastrojeni prema programerima, nisu se nezavisni proizvođači razvili ogroman broj dodatnih programa. Naime, oni su morali biti pisani u asembleru, tako da obični korisnici nisu bili u stanju da napišu svoje sopstvene programe. Osim toga, programi su pisani za određeni procesor, tako da nisu bili prenosivi. *Developer Tools* nije imao čak ni svoj editor o dibageru da i ne govorimo o a korisnik je morao konstiti sopstveni asemblar.

Poseban problem za programere je bilo ispravljanje grešaka, kao vrlo važan deo programiranja, koji čak nije bilo ni obrađeno u dokumentaciji.

Sada je situacija mnogo lakša, jer je na raspolaganju *Add In Toolkit* pomoću koga je relativno jednostavno napraviti dodatni program koji će savršeno raditi u saslavu sa 1-2-3 verzija 3.x. Zahvaljujući saradnji sa ekskluzivnim distributerom Lotusovih proizvoda na našem tržištu, M.P. YUSACO iz Novog Sada, u mogućnosti smo da prenesemo prve utiske o ovom razvojnom modulu.

Paket se sastoji od dve podebljane knjige. Obe su napisane visoko kvalitetno, što je uobičajeno za Lotusova upustva za upotrebu. Ukončena su patentiom u tvrde korice. To je kombinacija koju najviše volim, jer omogućuje lako listanje. Jedna knjiga je „Toolkit Reference i LPL Reference" a druga „1-2-3 R3 Library reference". Osim knjiga tu su, naravno, i diskete. Po dve veličine 5.25" i 3.5".

Instalacija je jednostavna i brza. Program *Linstall* otvara potrebne podirektorijume unutar 123R3 direktorijuma i u njih kopira dajloteke. Sve je to klasika, osim što program nije zaštićen. Kladim se da je jedan od razloga za to što u Lotusu ne smatraju našu i slične zemlje ozbiljnjim tržištem za ovaj proizvod.

ŠTA IMA U KUTIJI

Dakle, *Add In Toolkit Release 3.0* se sastoji iz nekoliko komponenti:

- Lotusov programski jezik (LPL)
- kompajler
- editor
- debugger
- biblioteka procedura

Kreiranje dodatnih programa zahteva nekoliko koraka:

1. upotreba editora i LPL za pisanje programa
2. kompilacija programa
3. pokretanje 1-2-3
4. učitavanje i startovanje programa unutar 1-2-3

Za korisnika OS/2 postoji pogodnost korišćenja dibagera pomoću kojeg je jednostavnije naći greške u programu i kongovati ih.

LPL je programski jezik specijalno napravljen da omogući proširenje mogućnosti Lotusovih proizvoda. Njegovim korišćenjem mogu se pisati modularni programi koji će raditi kao deo 1-2-3. LPL programi mogu biti sastavljeni iz više datoteka – modula koji će raditi kao celina, a naravno predviđene su i komande koje omogućuju kontrolu izvršenja toka programa.

Add In Toolkit sadrži „1-2-3 Release 3 Library". U toj biblioteci su tipovi podataka speci-

Listing 1 MENIPR

```
module mesipr is

Use Input, Cell

procedure Main ie

 begin
     string array[4] ShortPrompt, LongPrompt

     call ("poruka") answer

     ShortPrompt[1]= "Unos"
     ShortPrompt[2]= "Grafikon"
     ShortPrompt[3]= "Print"
     ShortPrompt[4]= "Kraj"

     LongPrompt[1] = "Unos podataka u tabelu "
     LongPrompt[2] = "Grafički prikas podataka "
     LongPrompt[3] = "Stampanje izveštaja "
     LongPrompt[4] = "Kraj rada "

  Menu (ShortPrompt  LongPrompt) Choices

     integer UserChoice = Input (Choices)

     case UserChoice

          of 1      : answer.value= "Korisnik je odabrao Unos."
          of 2      : answer value= "Korisnik je odabrao Grafikon."
          of 3      : answer value= "Korisnik je odabrao Print."
          of 4      : answer value= "Korisnik je završio rad."

     end case
 end
   except on others (string SignalName) do
     Print("Signal ", SignalName, " raised.")
     end except

end procedure Main
end module mesipr
```

Jednostavan i kratak primer pokazuje lakoću kreiranja sopstvenog menija koji će raditi kao deo 1-2-3

Listing 2 PUNJAC

```
module Punjac is

-- Format komande Punjac je :

--    (Punjac "Output",StartValue,StepValue,StopValue),

Use Register, Range

procedure Initialize is

     Register ("Punjac", Macro)
   except on others (string SignalName) do
     Print("Signal ", SignalName, " raised.")
     end except

     end procedure Initialize


procedure Punjac (string MyRange,
                  integer StartValue, StepValue, StopValue) is


  begin
        range[MyRange] Punjac
        if StartValue > StopValue then
     return
        end if
        erase(Punjac)
        float ThisValue    = StartValue
        for integer i = 1 to Punjac rows loop
                Punjac[i,1]  = ThisValue
                ThisValue = ThisValue + StepValue

                if ThisValue > StopValue then

     exit
                end if
        end loop
  end
    except on others (string SignalName) do
      Print("Signal ", SignalName, " raised.")
     end except
    end procedure Punjac

end module Punjac
```

Primer kako se može napisati program za novu makro komandu



tični za 1-2-3 (application data types – ADT) kao i procedure koje koristimo da omogućimo pristup podacima u 1-2-3. Na primer, za brisanje oblasti ćelija – *range* – u labeli iz LPL programa pozvaćemo proceduru **Erase** i saopštiti joj oblast. Ona će, pak, u našoj proceduri biti parametar tipa **Range**.

Procedure iz biblioteke procedura možemo koristiti u raznim situacijama. Pozivaćemo ih kada želimo

- manipulaciju ćelijama i oblastima
- kreiranje sopstvenih – namenskih menija
- izvođenje izlazno-ulaznih operacija na akranu
- kreiranje novih @ funkcija i makro komandi

U paketu je sadržan i editor. Mada nekog programiram i ne smatram se velikim znalcem, čini mi se da je sasvim dobar. U potpunosti je zadovoljio moje, za sada, skromne potrebe. Sadrži uglavnom sve neophodne mogućnosti koje nude i drugi dobri editori

- *cut* i *paste* kopiranje i štampanja
- *search* i *replace*
- rad sa više datoteka u memoriji, tako da se možemo kretati po tajlovima i kopirati delove teksta iz jednog u drugi
- startovanje kompajlera iz editora, tako da nakon prijavljivanja greške možemo odmah naslavljati sa editovanjem
- kontekstno osetljivi help koji „pokriva" i editor i dibager

Pre učitavanja i pokretanja dodatnog programa izvornu datoteku treba kompajlirati. Kompajler prevodi izvornu datoteku u LPL objektni kod i stvara objektnu datoteku. Kada zahtevamo izvršavanje programa, interpreter u 1-2-3 pokreće i izvršava objektnu datoteku.

Kompajler otkriva greške u izvornom kodu, prijavljuje ih i locira. Ako više nema grešaka, dobićemo poruku o uspešnoj kompilaciji. Nakon toga možemo učitati i pokrenuti naš dodatni program u okviru 1-2-3.

Ako koristite OS/2 verzija 1.1, onda možete koristiti i dibager koji će znatno pomoći u otkrivanju grešaka u programu. Ja nemam OS/2 i nisam mogao da testiram dibager, mada mi je žao što nije napravljeni za rad u DOS-u. Prema najnovijim informacijama iz Lotusa, nova verzija (3.1) *Add-In Toolkit* a će sadržati dibager za DOS.

Korisna adresa
M.P. YUSACO
21000 Novi Sad
Bulevar radničke samouprave 1
Tel. 021/395-414, Fax. 021/398 119

LAKOĆA PROGRAMIRANJA

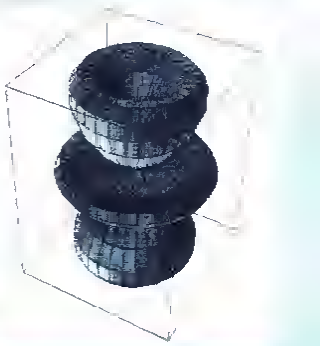
Sve ova komponente koje čine razvojni modul veoma olakšavaju rad na dodatnim programima za 1-2-3. Dakle, proces je jednostavan: editorom se piše izvorni kod u LPL, zatim se vrši kompilacija, a nakon toga možete konstiti vaš program kao sastavni deo 1-2-3. Obzirom na kratko vreme u koje imam na raspolaganju *Add In Toolkit Release 3.0* mogao sam se zabavljati samo na manjim primerima. Koristeći odličnu dokumentaciju i priložene primere, nije mi bilo teško da shvatim i prihvatim osnovne stvari u ovom razvojnom modulu. Uz ovaj članak idu i dva priloga. Na prvom je listing dodatnog programa sa konsničkim menijem (MENIPR). Program je jednostavan i kratak i demonstrira lakoću kreiranja sopstvenog menija koji će raditi kao deo 1-2-3. U drugom primeru (PUNJAC) se vidi kako se može napisati program za novu makro komandu. U ovom slučaju, radi se o makro komandi koja popunjava oblast ćelija brojevima sa zadatim početnim i krajnjim brojem, kao i korakom između njih. Oblast koju komanda popunjava brojevima može biti definisana bilo imenom bilo adresom.

Na sličan način možemo napraviti sopstvene @ funkcije ili kompletne aplikacije kojima će 1-2-3 služiti samo kao dobar i popularan interfejs. Obzirom da radim u termoelektranama „Nikola Tesla" gde se dobar deo poslova na PC računarima obavlja baš na *Lotusu 1-2-3*, zaceta je ideja da se napravi dodatni program za 1-2-3 koji će pomagati pri balansiranju velikih rotacionih uređaja, pre svih turbogeneratora. Realizaciju bi trebalo da vodi naš ekspert za te poslove *Miroljub Kovačević*, dipl. inž., a ako bude interesovanja, kod vas čitalaca, možda i objavimo to što uradimo.

U svakom slučaju, *Add In Toolkit Release 3.0* je veoma zanimljiv. Posebno onim programerima koji prave aplikacije za druge korisnike. Obzirom na rasprostranjenost i veliku upotrebu *Lotusa 1-2-3* u našim krajevima, mozda će se neko rešiti da pravi komercijalne dodatne programe. Tu će mu od neprocenjive koristi biti ovaj razvojni modul. U to i takve svrhe toplo ga preporučujemo.

MATHEMATICA 386

Nastavak sa strane 37



Milinko Stefanovic

ma inženjerskog posla u kome se ne barata sa integralnim i diferencijalnim računom. *Mathematica* u tom smislu čini izuzetnu softversku alatku. *Mathematica*-om se bez problema rešavaju kako određeni tako i neodređeni integrali. Da je proces rešavanja neodređenog integrala inverzna operacija diferenciranju, lako se može dokazati *Mathematica*-om. Jednostavno, rezultat integrisanja diferenciramo i dobijamo polaznu funkciju, a ovo pokazuje da je moguća i operacija diferenciranja. Međutim, poznato je da je proces rešavanja integrala mnogo teži nego proces diferenciranja, jer ne postoji jednistveno pravilo za njihovo rešavanje. Svakako da su definisani neki generalni principi, ali postoji vrlo velik broj integrala za koje su oni neupotrebljivi. Iz ovog proizilaze i (ne)mogućnosti *Mathematica*-e. Tako jednostavno se nalaze integrali racionalnih, trigonometrijskih ili eksponencijalnih funkcija, dok je *Mathematica* nemoćna pri rešavanju integrala tipa SIN(SIN(x)). Da bi se problem „nerešivih" integrala sveo na najmanju moguću meru, pre integrajenja poželjno je pojednostaviti funkcije nekim od metoda koje nudi ovaj softverski paket. Napomenimo da i kod operacija integrajenja i diferenciranja postoji mogućnost operisanja sa opštim brojevima.

Za one koji se bave eksperimentalnim radom a imaju potrebu da „fituju" rezultate eksperimenta, *Mathematica* pruža mogućnost da se ova operacija izvrši na više načina: linearnom, kvadratnom i eksponencijalnom funkcijom ili pak polinomom n-tog stepena. Za klasu problema koja se odnosi na eksponentne obezbeđena je i Furijeova transformacija i njena inverzna vrednost. U domenu problematike automatskog upravljanja, osobito stabilnosti sistema, vrlo je korisna mogućnost računanja Laplasovih transformacija. Na osnovu spiska specijalnih funkcija koje su implementirane u *Mathematica*-i evidentno je da su tvorci ovog softverskog paketa lakako vodili računa o kolegama iz svoje branše – teorijskim fizičarima. Data je mogućnost računanja Gamma, Riemann-ovih, Hurwitz-ovih, hipergeometrijskih i drugih funkcija koje se najčešće susreću u fizici.

Onim korisnicima *Mathematica*-e kojim nije dovoljno da konsla samo predvođene funkcije, dostupan je programski jezik *Mathematica*. Kao što smo u uvodu rekli, reč je o interpreteru, a predstavlja kombinaciju bejzika i LISP-a, uz napomenu da je programski jezik *Mathematica* ipak puno više nego puka kompilacija pretodna dva.

Nastavlja se na strani 41

JOŠ UVEK NA VRHU

Navikli smo da od svake nove verzije poznatog programa očekujemo mnogo. Imajući u vidu reputaciju koju ima editor *BRIEF*, od njegove nove verzije očekivalo se zaista puno, ali nas je prilično razočarala – osim podrške miša i nekoliko detalja, sve je ostalo po starom. Mnogo veće promene je donela nešto starija verzija 3.0.

Nenad Batoćanin

Izgleda da su miševi osvojili i zadnji bedem na svom putu koji je dosta dugo odolevao tom opsjem hrabrih editora teksta. Zašto je dosta programera uglavnom izbegavalo (i još uvek izbegava) korišćenje miša nije sasvim lako odgovoriti. Jedan od razloga koji se ne može baš ukusa je suvisno pokretanje ruke. Naime, pošto su editori vezani za kucanje po tastaturi nikako nije zgodno odvajati ruke sa tastature da bi se izvela neka operacija, pa zatim ponovo pomerati ruku na tastaturu. Međutim, naspram po sloje operacije koje je mnogo lakše uraditi mišem – na primer, pozicioniranje kursora na neki simbol koji nije blizu tekućoj poziciji kursora. U suštini miš je mnogo pogodniji za „grublje" operacije kod kojih nije potrebno opensati sa pojedinačnim redima. U novom *BRIEF*-u miša možete koristiti za pomeranje kursora markiranje dela teksta otvaranje i zatvaranje prozora itd. Kao dodatak mišu sad u *SETUP* u možete izabrati umesto uobičajenog okvira prozora i okvir na kome se vidi tekući položaj kursora u bonosu na početak/kraj datoteke i relativni položaj u tekstu. Može se izabrati i stari okvir ili potpuno isključiti svaki (što toplo preporučujemo)

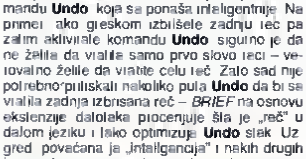
Korisnici *BRIEF*-a imaju još jedan razlog više da ne koriste miša. Naime, sve komande *BRIEF*-a su konciprane tako da se za njihovo aktiviranje olkuca što manje tastera, pa se normalne operacije obavljaju izuzetno brzo. Tako sasvim je sigurno da je za početak markiranja bloka lakše otkucati <Alt>m nego **bilo koju** kombinaciju tastera na misu. Naročito ako su ruke trenutno na tastaturi. Ipak, podška misa je svakako korak napred – ako vam i smeta ne morate ga koristiti.

Novi *BRIEF* podržava samo *Microsoft* kompatibilne miševa što na prvi pogled možda i nije nekog ograničenje ali izgleda da problemi ipak postoje. Naime, ni na jednom od tri računara (sa tri različita tipa misa) na kojima smo pokušali sa instalacijom *BRIEF* nije prepoznao misa! Očigledno da će drajveri za komunikaciju sa mišem morati da prerpe izmene.

Naravno, makro jeziku su dodate funkcije za rad sa mišem, a i dijalog menadžer (makro program koga *BRIEF* koristi za interakciju sa korisnikom) sad podržava polja koja treba „pokazati" misem.

UNDO I REDO

Undo je sigurno jedna od najmoćnijih komandi *BRIEF*-a, svaki prilsak na taster <Alt>u poništava efekat zadnje otkucane komande ma kakva ona bila. Ovakva komanda mnogo više pomaže u radu nego što izgleda na prvi pogled. Naime, *BRIEF* ima veoma moćna komande koje se dobijaju veoma kratkim nizovima tastera – skoro sve komande se aktiviraju prilskom na jedan taster. Takva mogućnost sa jedne strane omogucava izuzetno brzo manjanje teksta, a sa druge je veoma opasna – sa svega par tastera se može izbrisati cela datoteka! Ako imate na raspolaganju moćne komande, uvek se pomalo „bojite" da ne pogrešite i tako uništite ovoj trud, što rezulluje sporijim radom. Postojanje komande **Undo** vas oslobađa ovog straha i tako čini da bez bojazni upotrebljavate velike mogucnosti *BRIEF*-a. Nova verzija *BRIEF*-a ima donekle poboljšanu komandu **Undo** koja sa ponaša inteligentnije. Na primer, ako greškom izbrišete zadnju reč, pa zatim aktivirate komandu **Undo** sigurno je da ne želite da vratite samo prvo slovo reči – verovatno želite da vratite celu reč. Zato sad nije potrebno pritiskati nekoliko puta **Undo** da bi se vratila zadnja izbrisana reč – *BRIEF* na osnovu ekstenzije datoteka procenjuje šta je „reč" u dalom jeziku i tako optimizuje **Undo** slak. Uzgred povećana je „intailgancija" i nekih drugih komandi, na primer, komanda za kretanje po reč napred/nazad sada na osnovu ekstenzije datoteke termin reč shvata mnogo bolja. Na primer, u *C* programu je mnogo bolje da se **while()** shvati kao dve reči bez obzira što nema razmaka između.



Ono što je smešno kbo **Undo** komande je – sasvim neobično – velika brzina izvršavanja. Zaista, samo jedan prilsak više i može se nepovratno upropastiti veliki trud. Komanda koja rešava ovaj problem zove se **Redo**. Svaki pritisak na taster <Ctrl>u vraca zadnju **Undo** operaciju na **Undo** slak. Na primer, ako ste greškom izbrisali liniju, prilsak na **Undo** vraticе izbrisanu liniju. Sada će **Redo** učiniti suprotno, izbrisaće vraćenu liniju. Naravno, sad ponovo možete startovati **Undo** i tako ponovo vratiti izbrisanu liniju. Tako se možete „šetati" po **Undo** steku i izabrati stanje koje vam najviše bogovara. Kako lepo kaže u reklami za *BRIEF* sa **Undo** komandom možete slobodno eksperimentisati a **zaista** slobodno sa **Redo**.

MAKRO JEZIK...

Većina korisnika *BRIEF*-a ne koristi njegovo apsolutno najmocnije „oružje" makro jezik. Razlozi su jasni, ako vam programiranje nije „u krvi" teže ćete se odlučiti na pisanje programa za svaku, pa i najtrivijalniju stvar. Čak i okoreli profesionalci nace bas uvek posezali za pisanjem programa – nekad je lakše posao uraditi ručno ili običnim makroima. *BRIEF* je do verzije 3.0 imao tzv. tastaturni makro (*keystroke macro*) koji je omogucavao da zadate niz komandi i zatim ih ponovite proizvoljan broj puta. Ova lepa mogucnost je bila zaista korisna, ali sa puno mana: makro se uvek iznova morao definisati po svakom startovanju *BRIEF*-a, mogao se definisati samo jedan makro itd. Sada su ti problemi rešeni i sada možete istovremeno definisati samo jedan makro, ali možete ga snimiti na disk pod nekim imenom i kasnije učitati po potrebi – tako možete jednostavno napraviti zbirku korisnih rutina i po potrebi ih dodeliti pojedinim tastenma potpuno isto kao da ste pisali program u makro jeziku. Da stvar bude još lepša, makroi se snimaju u običnom ASCII formatu i to konsteći standardnu *BRIEF* notaciju za tastere, tako da su vrlo čitljivi i mogu se menjati u editoru. Što je izvanredno korisna mogućnost. Jedina mana u odnosu na makro programe je manja brzina izvršavanja što

Slika 1a

```
(macro primer
    |
        (int i)
        (= i 0)
        (beginning_of_line)
        (while ( i 10)
            |
                (insert "Tekst %d\n" |++ i|)
                (message "Linija broj %d" i)
            )
        )
    )
)
```

Primer programa pisanog na *Brief* makro jeziku

Slika 1b

```
primer ( )
{
    int   i;

    i = 0;
    beginning_of_line ();
    while (i  10)
        {
        insert ("Tekst %d\n", ++i);
        message ("Linija broj %d", i);
        }
}
```

Program na *CBRIEF* makro jeziku dobijen konverzijom *Brief* programa sa slike 1a



naročilo kod jednostavnijih operacija i nije kritično

Makro jezik je doživeo najveće promene. Naime, umesto ranijeg prilično nečitljivog makro jezika, uveden je *CBRIEF Macro Language*. To je pravi programski jezik sasvim blizak *C*-u, oni koji imaju bar malo iskustva sa *C*-om neće imati problema sa pisanjem u ovom jeziku. Praktično jedina razlika je u funkcijama koje su specifične za *BRIEF*, dok su kontrolne strukture potpuno iste kao u *C*-u. Makro jezik se razlikuje samo na *source* nivou, dok je prevod (datoteka sa ekstenzijom CM) ista kao ranije. To znači da će svi vaši stari makroi raditi bez problema. Šilo se liče pisanja novih makro programa možete birati način pisanja jer postoje dve verzije prevodioca: jedan koji prevodi *CBRIEF Macro Language* i drugi koji prihvata dosadašnji *BRIEF Macro Language*. Postoji i konvertor programa iz jednog u drugi oblik. Na slici 1a i 1b prikazani su primer makro programa i rezultat rada konvertora.

...I MAKRO DIBAGER

Sasvim logičan nastavak koncepcije *BRIEF* makro jezika kao pravog programskog jezika je *debugger*. Ništa neobično jer makro programi mogu biti složeni kao i svi „normalni" programi, pa je otkrivanje grešaka bez *debugger*a bilo veoma otežano naročito zbog specifičnog položaja makro jezika – program se izvršava u drugom programu (editoru) i ne može se posebno izdvojiti. Sada postoje dve verzije *BRIEF*-a koje su spolja identične. Jedan se startuje komandom **b «lista datoteka»** a drugi komandom **bd «lista»**. Po startovanju se na ekranu ne uočava nikakva razlika i sve komande normalno rade. Međutim, ako ste startovali editor bd i ako otkucate **debug** pojaviće se meska sa sinkom 2. O komandama se može reći da su prilično kompletna tu su standardne komande za izvršavanja programa (makroa) naredbu po naredbu, postavljanje prekidnih tačaka uslovnih prekidnih tačaka po nekoj (jednostavnijoj) relaciji, pregled i izmena vrednosti promenljivih itd. Ipak, rad sa *debugger*om je mogao biti i jednostavniji. Ovako morate prvo da startujete specijalnu verziju *BRIEF*-a pa *debugger* pa da učitate makro... Neka od koncepcija u kojima je editor integrisan sa *debugger*om bi mnogo više pristajala. Uostalom *BRIEF* i jeste editor – zašto programiranje u njemu onda razdvajati od *debugger*a?

OSTALE NOVINE

Podrška EMS memorije je lakoće veoma značajna novina. Ranije je EMS memorija korišćena samo pri izvršavanju DOS komandi – tada se jedan deo *BRIEF*-a premeštao u EMS, oslobađajući tako oko 150k osnovne memorije. Sada se EMS koristi za čuvanje datoteka i makroa što bi trebalo značajno da ubrza rad na sistemima sa većom EMS memorijom jer sa lako isključuje potreba za čestim prisupom disku.

Ostalo su manje-više sitne promene: statusne linija sada može da ispisuje 80 simbola, dijalog menadžer podržava još neka vrsta izbora, napravljena je podrška za neke prevodioce itd. Ono što je možda interesantnije za korisnike *Microsoft C 6.0* urađena je podrška za *Microsoft Programmer's Workbench* i *Microsoft Quick Help*.

Da li se znali da datoteke iz *BRIEF*-a može da birate i iz menija? Ako ste malo pažljiviji čitali uputstvo znacete da je ova mogućnost postojala i u stanjim verzijema od 3.0. Ova mogućnost je vrlo praktična. Na primer, ako vam treba da učitate neku datoteku iz takućeg kataloga a zaboravili ste kako se zove možete je učitati i ako znate samo deo njenog imena. Otkucajte «Alt-e» (komanda za učitavanje nove datoteke) otkucajte konščenjem džoker simbola ime datoteke koju želite da učitate (na primer **pr*.for**) i pritisnite taster «Tab». Ako u tekućem katalogu postoji samo jedna datoteka koja zadovoljava zadati uslov ispisaće se ime te datoteke i vama preostaje samo da pritisnete «Enter» kao potvrdu za učitavanje. Ako postoji više datoteka ispisaće se meni koji sadrži spisak svih datoteka, iz koga možete izabrati onu koja vam treba.

Sasvim je sigurno da je *BRIEF* još uvek najbolji izbor u klasi profesionalnih editora. Iako na postoje neke mane koje se dosta dugo provlače pa kako nisu otklonjene ni u novoj verziji očigledno je da autori *BRIEF*-a na namarivaju skoro da odstupe od svoje početne linije: polpuna otvorenost za izmene. Naime, lepo je imati dijalog menadžer u tekstu, potpuno otvoren za izmene, ali je možda bolje na najboljna ki bolji kompromis između zatvorenog ali brzog i sporog ali otvorenog konsnićkog interfejsa. Ove loša osobina se sasvim lepo vidi kad se uporedi *BRIEF* help sistem sa sistemom na primer *Turbo C*-a. Jednostavni meniji u *BRIEF*-u izgledaju vrlo neugledno i jadno pra ma moćnom hiperteksi sistemu u kome su sve informacije kao na dlanu.

Takođe mogli bi se upotpuniti jezički pakeli. Do sada je najmoćnija podrška raznim *C* prevodiocima. Šlo je dobro ali ne bi smetala isla lakva podrška (dopunjavanje naredbi ira ženje procedura komentarisanje blokova) i za ostale jezike.

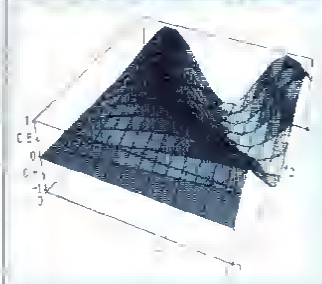
Redefinisanje tastature je zaista izuzalno kompletno ali morao bi postojati nekl atraktivniji program za mapiranje kompletne tastature od **keys** makroa.

Ipak, po sasvim subjektivnom sudu, novi *BRIEF* još uvek ostaje u samom vrhu profesionalnog softvera na MS-DOS-u. Kako kaže njegovo uputstvo za korisenje lokus *BRIEF* koncepcija bio je i ostao usmeren na tri zahteva brzina, moć i sigurnost. A koga nikad dosta bas kao brzine procesora i veličine memorije kod računara.

Zahvaljujemo Institutu za računarsku tehniku i telekomunikacije iz Vinče što nam je za potrebe ovog lesta slavila softver na raspolaganje.

MATHEMATICA 386

Nastavak se strane 39



Ova razlika se posebno odnosi na samu filozofiju programiranje jer euton preporučuju da se porgramen sa programetskim iskustvom u drugim programskim jezicima (npr fortranu), oslobode „negativnog nasleđa" i prihveta koncepta koji nudi *Mathematica*. Ovo nije imperativ ali kako se navodi u uputstvu, za matematičke probleme treba koristiti upravo njihov interpreter. Auton ovog softvera daju i vrlo zanimljivo tumačenja zašto se *Mathamatica* opredelila ze interpreter. Preme njihovim saznanjima koja su neosporno velika vrama koja se utroši na izvrašavanju programa zanemarljivo ja malo u odnosu na vrema utrošeno na njegovom razvoju. Dakle neophodno se prilagoditi što efikasnijem tazvoju softvera. Jasno je, takođe da se kompilirani kodovi brže izvršavaju ali ta sa prednosi gubi u odnosu na interpreter ako se primenjuju algoritmi kakve nudi *Mathematica*. No, ako ipak konsnik ovog paketa ne žali da se oslobodi svojeg programerskog „nasleđe" moguća je direktna translormecija koda napisanog u *Mathematica*-i u kod fortrana ili C-a.

Sasvim optavdano pitanje se postavalja kako sve prezentoveti rezultate dobjene u radu sa ovim softverskim paketom. Za ovo postoji više načina. Rezultati samog proračuna formula ili pak izvorni kodmogu se poslati u odgovarajuće datoteke i potom štampati na specificiram štampač. Međutim mnogo veće zadovoljstvo pružeju rezultati koje *Mathematica* daje u grafičkoj formi. Kao što smo rekli u uvodu moguće je prikazivali lunkcije u 2-D i 3-D prostoru na najrazličije načine. Grafička eplikacija se dobija direktno iz programe ili preko eksterne komande iz MS-DOS-a. Za aplikacije koje sa prikazuju iz MS-DOS-a na ekranu računara, neophodno je prethodno formirati datoteku u *PostScript* formatu koja se učitava komandom DISPLAY. Komandom Hardcopy, ako je u pitanju *PostScript* printer, slika u ovom formatu se direkino šalje na printar. Međutim, ako je u pitanju neki drugi printer komandom HARDCOPY poziva se eksterni program PRINTPS koji konvertuje formiram *PostScript* datoteku u bitmapiranu i šalje je ka printeru.

UMESTO ZAKLJUČKA

Obim i mogućnosti softverskog paketa *Mathematica* su takvi da ih je nemoguće predstaviti u jednom članku. Pre svega, radi se o paketu koji svoje najveće vrednosti pokazuje u domenu „više" matematike. U ovom članku je zalo bilo reči samo o jednom delu *Mathematica*-e. Preciznije da bi se shvatilo šta se sve može tražiti sa ovim softvetskim paketom treba ga jednostavno kupiti i uživati u njemu. Koma je matematike deo svekodnevne profesionalne aktivnosti neće se pokajati.

PROBAJTE SLOBODNO

I pored toga što se na tržištu nudi veoma mnogo programerskih editora, od proslavljenih *Brief* i *Epsilon* do editora koji dolaze u integrisanom okruženju uz gotovo svaki ozbiljniji kompajler, programeri imaju razloga da dobro razmisle šta će odabrati. Jedan od editora koji zaslužuje punu pažnju je i *Qedit*.

Pavle Pekić

Giganti među programerskim editorima kao što je recimo *Brief* imaju mnogo aduta kojima osvajaju programere širom sveta ali i jednu značajnu manu koja u velikom broju slučajeva odbija programere da se odluče baš za njih - strahovito su glomazni. Dodavanje moćnih opcija je išlo na uštrb jednostavnosti upotrebe kao i na brzinu rada sa samim editorom. Pogledajte samo mnoge od čuvenih editora, koliko im samo vremena treba da se pokrenu!

PRAVO U EDITOR

Upravo su tu ljudi iz SemWare-a, autora *Qedit*a, videli svoju šansu i pokušali da objedine mnoštvo opcija sa jednostavnošću i brzinom rada. Videćemo koliko su u tome uspeli.

Qedit je *shareware* proizvod. Sve što je potrebno za njegovu osnovnu instalaciju je da se oformi direktorijum (npr *qedit*) i da se u njega izvrši raspakivanje datoteka. Editor je odmah spreman za rad i možemo ga startovati sa

O <ime fajla>

Naravno, ime fajla koji želimo da editujemo možemo izostaviti u kom slučaju će nam *Qedit* prikazati na ekranu sadržaj tekućeg direktorijuma složenog u jednu standardnu listu za izbor. Dužina editovanog fajla je ograničena veličinom slobodne memorije. Na žalost, *Qedit* nema mogućnost korišćenja *expanded* ili *extended* memorije u tu svrhu, iako ume da koristi proširanu memoriju za *swap* prilikom izlaska u DOS. Verovatno iz ovog razloga, programeri SemWare a su se potrudili da naprave što kraći program, iako da *Qedit* ima svega nešto manja od 50K, što znači da u proseku za editovanje ostaje preko 500Kl. Takođe, za rad *Qedit*-a je dovoljan samo jedna jedina daloteka – Q.EXE. Sve promene prilikom konfigurisanja ili bilo kakvih drugih izmena se direktno upisuju u program.

Qedit sigurno ne bi bio vredan pažnje da nema mogućnost jednostavnog istovremenog rada sa više fajlova. To se postiže na više načina. Moguće je jednostavno iz komandne linije napisati

O <ime1> <ime2> ... <imen>

i *Qedit* će učitati željena daloteka, naravno ako im ukupna dužina ne prelazi veličinu raspoložive osnovne memorije u kom slučaju će biti učitano samo onoliko fajlova koliko može stati u slobodnu memoriju. Takođe u svakom trenutku je moguće učitati novi fajl u prsten fajlova, kako su ga u SemWare-u nazvali, ili pak izbaciti neki iz prstena. Kretanje kroz prsten se može obavljati u oba pravca. Ij. napred ili nazad, a postoji i mogućnost navođenja imena fajla u prstenu koji trenutno želimo da editujemo.

Više fajlova možete učitati i korišćenjem džoker znakova – dozvoljeno je napisati sledeće

O *.doc

i *Qedit* će učitati sve datoteke iz tekućeg direktorijuma sa ekstenzijom .doc.

Pošto smo već opisali rad sa više fajlova, treba pomenuti da je na ekranu u isto vreme moguće i videti dva ili više dokumenata, svaki u svom prozoru. Za rad sa prozorima postoje sve standardne procedure na koje smo navikli i u ostalim editorima, kao što su zumiranje, promena veličine, zatvaranje prozora i slično. Ovde smo primetili jedan sitan propust – kada na VGA ili EGA karti podelimo ekran u modu sa 25 redova u dva dela, a zatim pređemo u mod sa 43 reda, opet ćemo dobiti samo jedan prozor na ekranu.

REDEFINISANJE TASTATURE

Nakon startovanja editora i izbora fajla/fajlova za editovanje naći ćemo se u samom srcu editora. Ij. u njegovom osnovnom ekranu na kome će nas dočekati izabrani fajl i statusna linija koja sadrži informacije o redu/koloni u kojoj se nalazi kursor, indikatore za insert, automatsko uvlačenje (*autoident*), tip preloma u novi red (*wordwrap*), snimanje makroa (*macro recording*), crtanje okvira (box drawing), kao i veličinu slobodne memorije. Ime fajla koji se edituje i indikator promene fajla. F1 prikazuje kratak HELP za komande. Upravo je HELP najslabiji deo editora, jer pritisak na F1 daje uvek samo raspored komandi i ništa više. Ali da budemo precizniji, F1 daje na ekran sadržaj Help fajla, koji je najobičniji ASCII fajl i koga možemo u bilo kom trenutku, čak i sa *Qedit*-om, izmeniti i pokrenuti konfiguracioni fajl da bi se njegov sadržaj prepravio u program. No, ne treba se previše brinuti zbog nedostatka dobrog HELP sistema, jer je za *Qedit* na raspolaganju *Norton Guides* baza.

Ako nam kojim slučajem trenutno definisane funkcije tastera ne odgovaraju, *Qedit* nam omogućuje da biramo između nekoliko emulacija rasporeda čuvenih editora i tekst procesora, kao što su *Norton editor, Brief, WordStar, WordPerfect* i drugi. Na kraju, ako vam čak ni jedna od ovih emulacija ne odgovara, što je slučaj sa autorom ovog članka, imate jednostavnu mogućnost kreiranja sopstvenog rasporeda funkcija editora na tasteru. Definicija tastera se nalazi u jednom ASCII fajlu, tako da za njegovu izmenu možete koristiti i sam *Qedit*. Procedura je veoma jednostavna i svodi se na ispisivanje imena tastera, kao npr

f1, ^f1 (SHIFT+F1), ^F1 (CTRL+F1) i al

a u produžetku je potrebno navesti identifikacionu reč za željenu funkciju, na primer

f1	QuickHelp
f2	SaveFile
l3	File
f4	Find
f5	RepeatFind
l6	FindReplace
l7	GotoLine
l8	UnmarkBlock

Za konfigurisanje ca vam od velika pomoći biti *Norton Guide* za *Qedit* u koma imate listu svih funkcija i identifikacionih reči za svaku od njih. Konfiguracioni fajl ne nosi u sebi samo dodele funkcija editora tasterima, već i makro definicije. Kao što je tasteru f7 dodeljena funkcija skoka na željenu liniju, **GotoLine**, možemo mu dodeliti i više funkcija

f7 MacroBegin MarkBlockEnd Cut

Po završenom redefinisanju tastature, po novićemo standardnu konfiguracionu proceduru i naša nova redefinicija tastature nam je na raspolaganju.

BLOKOVI I PRETRAŽIVANJE

Rad sa blokovima je jedan od najvažnijih testova kvalitela nekog editora. Autori *Qedit*a su to izgleda shvatili i evo šta su nam omogućili za manipulaciju sa blokovima:

Tabela 1

Clipper by Nantucket 5/87	prg	"clipper	$N	>$O" line „
FoxBase Compiler v2.1	prg	"foxpcomp -e	$F	>$O","Error in line „
FST Modula-2 v2.0a	mod	"m2comp	$F	>$O","File $F", Line"
Microsoft Assembler 5.x	asm	masm	$F	>$O
Microsoft BASIC	bas	bc /Z	$F	>$O
Microsoft C 5.x	c	cl /c /W3	$F	>$O
Microsoft FORTRAN 4.0	for	fl /c	$F	>$O
Microsoft Manu Maker v1.2	def	"makemanu	$N	>$O","Error („
Microsoft Pascal 4.0	pas	pas1 /H	$F	>$Q
Alternative	pas	pl /c /Zz	$F	>$O
Microsoft Quick C 1.01	c	qcl	$F	>$O
QMac (QEdit macros) 2.x	qm	"qmac $N mac	$F	/B /A- >$O"
„QMAC „ Telix 3.x Script Compiler	sll	cs	$F	>$O
TopSpeed Modula-2 1.14	mod	m2 /c	$F	>$O
Turbo Assembler 1.00	asm	"tasm	$F	>$O" ,** $F"
Turbo C 2.0	c	tcc -c -w	$F	>$O
Turbo Pascal 4.0 & 5.0	pas	tpc /q	$F	>$O
Watcom C 6.5	c	wcl /c	$F	>$Q

Obeležavanje dela teksta je moguca na tri nacina. Možemo obeležiti jedan red, niz redova kao i isacak teksta. Takode ovo je moguce ostvariti pomoću dva sistema markiranja. Jedan je klasican, kakvog poznajemo još iz vremena prvih verzija *WordStar*-a, u kojem je potrebno postaviti marker za početak a zatim otici do kraja željenog bloka i postaviti marker za njegov kraj. Naravno, marker kraja mora biti posle markera početka bloka, a ceo blok na ekranu vidimo obeležen i spreman za manipulaciju tek kada su oba markera postavljena. Na sreću, postoji i drugi, mnogo ticksibiliniji jednostavniji i brži način markiranja – postavljanje znakova za obeležavanje. Kada se postavi jedan znak za obelezavanje u isto vreme se drugi vezuje za kursor – kako pomerate kursor lako se blok širi ili skuplja. Naravno, sasvim je svejedno koji marker označava kraj bloka a koji početak – to se određuja automatski u zavisnosti koji je ispred a koji iza. Brzina ovakvog rada se ogleda u tome što nije potrebno lako je moguce tiksirati drugi marker. *Qedit* računa da ja drugi marker na tranutnoj poziciji kursora.

Bez obzira koju metodu označavanja upotrebili definisani blok ja moguca izdvojiti (**Cut**) i preneti u bafer za operaciju spajanja (Paste) koju naravno možemo i visestruko izvršavati sa istim sadržajem batera. Takode moguca je i kopiranje i brisanje označenog bloka kao i **copy_over** i **paste over** funkcije. Zanimljiva su još i **shift_left** i **shift right** funkcije koje pomeraju sadrzaj bloka za jedno polje ulevo ili udesno. Po završetku željene manipulacije sa označenim blokom skinućemo mu markera komandom **unmark**.

Standardan deo svakog editora su i funkcije za traženje i zamenu dela teksta. Zadati deo teksta možemo tražiti na više načina: od kursora do kraja fajla, ili od kraja dokumenta unazad. Moguće je tražiti deo reči ali i samo cele reči, sa uvažavanjem razlike izmedu malih i velikih slova i ignorisanjem tog uslova – sve to kroz ceo dokument ili samo kroz izdvojeni blok. Logično, tu je i funkcija za nastavak traženja zadatog dela teksta koja, u slučaju da pre nje nije upotrebljena **find** funkcija, poziva tu funkciju.

Zamena jednog dela teksta drugim sadrži identične mogućnosti kao i funkcija pretraživanja, s tim što sadrzi i opciju kojom se odreduje da li će se sve nadena reci automatski zameniti ili će se pre svake zamene tražiti dozvola od korisnika.

Ipak, nedostajala nam je mogućnost pretraživanje sa korišćenjem dzoker znakova ili čak logičkih izraza.

Qedit naravno omogućuje i štampanje celog ili dela teksta, kao i postavljanje osnovnih parametara štampanja: leve, desna, gornje i donje margine, kao i dužine stranice.

Tu su i sva standardne funkcije za editovanje, kao i veoma važna UNDO funcija za vracanja teksta izbrisanog pomocu komandi za brisanje bloka, linija, reči ili za brisanje teksta od kursora do kraja reda. Veličinu batera za čuvanje izbrisanih delova teksta odreduje sam korisnik, a maksimalna velicina je 3000 redova.

DVE ZNAČAJNE OPCIJE

Qedit pored svih pomenutih funkcija sadrži i mnogo mnogo drugih funkcija koje olakšavaju rad, kao i prekidača za odredene modove. Nabrojati i objesniti sve bi se svelo na pisanje priručnika a ne prikaza editora. Ipak, ostale su nam još dve vazne oblasti koje smo namerno ostavili za kraj, jer želimo da im poklonimo posebnu pažnju. A to su makroi i povezivanje *Qedit*-a sa kompajlerima.



Kreiranja makroa je moguće na dva nacina – editovanjem konfiguracionog fajla za redefinisanje tastature i dodeljivanjem makroa željenim tastenima, kao i interaktivno tokom rada. Prvi način smo već opisali kada smo govorili o dodeljivanju funkcija tasterima. Drugi način je veoma jednostavan i svodi se na izbor opcije **MacroRecord**, nakon koje izaberete taster kojem zelite da dodelite makro. Sve što dalje bu dete uradili bice prenoseno kao makro sekvenca koju završavate ponovnim izborom opcije **MacroRecord**. Makroe možete snimiti i kasnije ih ucitati u zavisnosti koji vam u datom trenutku treba. *Qedit* omogucuje definisanje nekih osnovnih parametara za razlicite tipove dokumenata, ali to u mnogim slučajevima nije dovoljno. Zato sa mogu jednostavno definisati makroi koji će se pozvati u zavisnosti od tipa teksta koji adilujemo i postaviti labove, margine i druge važne opcije.

Ipak najznačajnije je što uz pomoć makroa i eksternih programa QCP i QMAC možete povezati *Qedit* sa većinom kompajlera. Kako?

QCP ili *Qedit compiler* je takode proizvod SemWare a a istnburn se kao zaseban paket. U tom paketu, pored samog programa i dokumentacije, postoji i makro fajl sa definisanim makroima za pozivanje QCP-a iz *Qedit*a. *Qedit* prosleđuje ime i ekstenziju fajla, kao i kompletan putokaz QCP-u, a QCP na osnovu ekstenzije teksta poziva odgovarajuci kompajler sa željenim parametrima, preusmeravajuci izlaz kompajlera umesto na ekran u fajl na primer ERRORS.LST. Podaci o ekstenzijama fajlova, imenu programa za poziv kompajlera kao i opcijama kompajlera i opcijama o strukturi greške nalaze se u datoteci QCP.DAT koja je kao i svi ostali konfiguracioni fajlovi u *Qedit*-u obicen ASCII tekst. Struktura poziva kompajlera u odnosu na ekstenziju izgleda ovako:

EXT '<**kompajler**> <**parametri**>" "string1" "string2"

gde EXT predstavlja ekstenziju datoteke, <kompajler> ime kompajlera koji se poziva a <parametri> prekidače sa kojima će biti pozvan kompajler. *string1* je tekst koji se pojavljuje uz svaku grešku tj. jedinstvena rec na osnovu koje QCP detektuje da se u tom redu nalazi greška, kao i poruka uz tu grešku. *string2* je tekst koji sa uvek nalazi levo od broja reda u izvornom kodu u kome je greska. Parametri string1 i string2 se mogu izostaviti.

QCP omogućuje povezivanje sa većinom poznatijih kompajlera, što znači ipak ne sa svakim, iz prostog razloga što su vece kuce proizvodači kompajlera usvojili neke nazovimo ga standard izlaznog izveštaja kompajlera. QCP na osnovu tih nekih standardnih oblika ispisa pronalazi broj reda i poruku o greški u tom redu u izlaznom fajlu ERRORS.LST.

Da vidimo kako taj usvojeni standardni ispis izgleda tj. kako izgledaja struktura izlaza greška koju kompajler prijavljuje:

<**text**> **FILENAME EXT** <**nenumericki znaci**> **XX** <**tekst**> **YY** <**poruka**>

gde je XX broj reda u tekstu programa u kome se pojavila greska, a YY je broj kolona u tom redu. Dakle, na izlazu se mogu pojaviti sledeci oblici:

FILENAME.EXT (RED KOLONA) PORUKA
FILENAME.EXT RED PORUKA
„FILENAME.EXT RED KOLONA PORUKA
D:\PATH\FILENAME.EXT(RED) PORUKA

Da bi sve ovo bilo mnogo jasnije primere poziva nekih poznatijih kompajlera dajemo u tabeli 1.

Ako se pri kompajliranju pojavi greška i QCP detektuje brojeve i nju i poruka o greška ma, on poziva QMAC (*Qedit Macro Compiler*) koji makroa za laženje redova u izvornom kodu kojima se pojavila greška kompajlira u bi nami oblik za upotrebu u *Qedit*-u.

Po završenom poslu, upravljanje se vraca *Qedit*-u i to lako što se ekran podeli u dva prozora – u gornjem se nalazi tekst programa a u donjem poruke o greškama. Sada vam samo ostaje da ispravite greške u vasem programu, što ce vam olakšati definisani makroi. Ta starom SHIFT-F9 *Qedit* će obaleziti sledecu gresku u prozoru za greske i pronaci odgovarajući red u prozoru sa tekstom programa. Kombinacijom SHIFT F10 se skače na prethodnu grešku. Naravno, ove tastere kao i sve makroe moguce je po zelji promeniti i prilagoditi sopstvenim potrebama.

Qvakav način kompajliranja iz editora smanjuje vreme programiranja i umnogome olaksava otklanjanje gresaka. Programi QCP i QMAC zajedno sa *Qedit* om predstavljaju zaista moćno oruđje u kreiranju programa i svakom programeru koji sa još nije definitivno opredelio za neki od editora savetujemo da obavezno razmotri i *Qedit*, jer ima puno loga da pruži.

Registracija programa

Qedit je *shareware* program kuca SemWare. Ako vam se u probnom periodu *Qedit* dopadne i požalite da sa registrujete uplatićete iznos od 54.95$ + 10$ manipulativnih troškova za verziju sa stampanim priručnikom, ili 44.00$ + 5$ za verziju se priručnikom na diskeli. Kuća SemWare daje kompletnu podršku telefonom ili putem sopstvenog BBS-a.

SemWare 4343
Shallowford Rd. Suite C-3
Marietta, GA 30062-5003

ili kreditnom karticom VISA/MC/AMEX

Tel. (404) 641 9002
BBS (404) 641-8968
Fax (404) 640-6213

Uzmite sa Sezama

Sve opisane programe i datoteke (*Qedit*, QCP, QMAC, emulacije rasporeda tastera kao i Norton Guide za *Qedit*) možete naći na Sezam BBS-u (648-899) u direktorijumu **\Ibmpc\editor** a za sva eventualna pitanja i probleme u sa vazi *Qedit*-om ostavite poruku autoru na pseudonim **ppekovic**.

GEOMETRIJSKO MODELIRANJE

Osnovna razlika između amatera i profesionalnog korisnika CAD paketa je u tome što prvi poznaje odgovarajucu komandu i šta ona radi, dok drugi treba da zna i kako ona radi. Pokušaćemo da vam pomognemo u prelasku iz prve u drugu kategoriju korisnika. To izgleda ovako...

Slavko Kerač i Milan Bojanović

Na osnovu dosadašnjih članaka iz serije projektovanje pomoću računara jasno se može zaključiti da metodi geometrijskog modeliranja predstavljaju sinlezu više naučnih i tehničkih disciplina kao što su na primer analitička i nacrtna geometrija topološka teorija skupova numerička analiza vektorski i matrični račun i slično

DA PROBAM PA ŠTA ISPADNE

Cilj detaljnijeg upoznavanja sa „tehnologijom" geometrijskog modeliranja odnosila pomoću krivih i površi je da se softverski paketi koji se bave ovom problematikom ne posmatraju kao „crne kutije" koje rade po modelima rezervisanim za dobre poznavaoce matematika Takode upoznavanje sa matematičkim modelima krivih i površi koje se koriste u CAD u treba da smanji broj korisnika CAD softvera koji rade metodom „probaću pa šta ispadne" Cilj je upravo obrnut stvoriti obrazovanog projektanta koji će unapred znati kakav će oblik, npr vetrobranskog stakla automobila dobiti zavisno do tipa površi kojom je opisao taj objekat Precizuje na primer profesionalni korisnik *AutoCAD*-a treba da zna unapred kakav oblik površi treba da očekuje ako koristi SURFTYPE 6 (kubne B splajn površi) a kakav ako koristi SURFTYPE 8 (Bezierove površi)

Ključne elemente u geometrijskom modeliranju objekata predstavljaju krive i površi pri čemu postoje dva načina njihova upotrebe u procesu modeliranja **analitička upotreba** i **aintetička upotreba** Analitički pristup podrazumeva da kombinacijom raznih analitičkih krivih i površi opisujemo vec postojeci objekat da bismo izvršili odgovarajuce proračune ili modifikacije na njemu Sintetički pristup podrazumeva da model objekta moramo izgraditi što znači da on još ne postoji Očigledno je da ova dva načina zahtevaju različite projektna postupka Objasnimo ih detaljnije

Obično upotrebljavamo dva nacina za predstavljanja površi mnogougaonu mrazu („polygon mesh") i parametarske segmente površi („parametric patches") Mreža je skup pojedinačnih planarnih mnogougaonika koji ograničavaju neki objekat dok pod parametarskim segmentima površi podrazumevamo površi ograničena sa prostornim parametarskim krivim

U procesu geometrijskog modeliranja postupci predstavljanja krivih i površi moraju imati sledeće osobine

1 kontrolne tacke

Uobičajen način interaktivnog upravljanja oblikom krive je ili postavljanje tačaka kroz ko je kriva mora da prode ili postavljanje tacaka koje ulicu na oblik krive Te tačke su kontrolne tačke Ako kriva prolazi kroz kontrolne tačka kažemo da kriva interpolira kontrolne tačka

2. visestruke vrednosti

Pri određenoj nezavisno promenljivoj, kriva obično ima više vrednosti

3 nezavisnost od koordinatnog sistema

Ako kontrolne tačke definišemo u različitim koordinatnim sistemima oblik krive se ne sme menjati

4 lokalna kontrola

U slučaju kada se sa promenom položaja jedna kontrolne tačke promeni kriva samo u blizini te tačke Imamo lokalnu kontrolu

5. osobina smanjivanja varijacije („variation-diminishing property")

Ova osobina gladi krivu oko kontrolnih tačaka i onemogućava oscilirajući oblik kriva

6 raznolikost („versatility)

Predstavljanje krivih mora da dozvoli pravljenje najrazličijih oblika

7 red neprekidnosti sastavljenih krivih

Komplikovanje oblike obično ne modeliramo samo sa jednom krivom nego uz pomoć više sastavljen-h krivih Taj postupak koji u dodirnim tačama sastavljenih krivih često uvodi neželjene oštre prelaze, nazivamo **prustup po segmentima** („piecewise approach") Želja projektanta je da moze ulicati na red neprekidnosti u tim dodirnim tačkama U dodirnoj tački često zalimo da imamo neprekidnost prvog re da (C^1) slo znači jednake vektore prvih izvoda ili neprekidnost drugog reda (C^2) što znači jed naka vektore drugih izvoda

Zahtevane osobine najlakse ispunjavamo ako krivu ili površ zadamo u parametarskom obliku Razlozi za ovakav prslup bice objašnjeni u naslavku

DEFINISANJE KRIVIN LINIJA I POVRŠI

Geometrijske elemente možemo matematički predstaviti pomoću parametarskih ili nepa rametarskih jednačina Neparametarske jednačine mogu biti u eksplicitnom i implicitnom obli ku

Kriva u ravni u eksplicitnoj neparametarskoj formi mogu se predstaviti u generalnoj formi

$$y = f(x)$$

Za krivu opisanu ovakvom jednačinom postoji samo jedno **y** za svako **x** Eksplicitna forma može predstavljati zatvorene ili visezačno određene krive Ova ograničenja mogu se prevazići koriscenjem implicitnih jednači na u generalnoj formi

$$f(x\ y) = 0$$

I implicitni i eksplicitni metod su zavisni od koordinatnog sistema slo je često nepozeljno mada se navedena metode u nekim aplikacija ma mogu vrlo uspešno koristiti Druga teskoca nastaje kada se koriste koordinatno zavisne neparamatarske krive i kada krajnje tačke na krivoj imaju nagib od 90 u odnosu na izabrani koordinatni sistem (salila se problema kako predstaviti tangens ugla od 90 stepeni) Tada je nemoguce koristiti beskonačni nagib koji karakteriše te tačke kao granicni uslov U ovom slucaju koordinatni sistem se mora promeniti ili se mora vrsiti aproksimacija beskonačnog nagiba sa velikim pozitivnim ili negativnim vrednostima

U geometrijskom modeliranju elemenata skoro da je nemoguće prikazati neki objekat samo sa jednačinom oblika $y = f(x)$ Po pravilu uvek se zeli modelirati objekat koji je nezavisan od koordinatnog sistema Ovo znači da je u toku modeliranja uvek bitan odnos izmedu tačaka koja delimišu objekat a ne izmedu tih tacaka i proizvoljno odabranog koordinatnog sistema Prema toma vecina aplikacija zahteva da izbor koordinatnog sistema ne utiče na oblik modeliranog objekta Takode evidentna je činjenica da bilo koji čvrsti odjekat ima vertikalne tangentna linija ili ravni u odnosu na bilo koji izabrani koordinatni sistem Ovo razuliuje beskonačnim nagibom ili loša delimisanim matematičkim karakteristikama Konačno krive i površi u geometrijskom modeliranju često ne leze u ravni, već su delimisane u prostoru pa ne mogu uopšte biti predstavljene obicnim neparametarskim funkcijama

Zbog navedenih razloga kao i mnogih drugih koji se odnose na mogućnosi računarskog modeliranja dominantno mesto u geometrijskom modeliranju pripada parametarskim jednačinama Na primer dvodimenzionalna kriva se na predstavlja u obliku $y = f(x)$ vec kao sistem od dve funkcija $x = x(u)$ $y = y(u)$ gde su **x** i **y** funkcije parametra **u** Tačka koja pripada lakvoj krivoj predstavlja se vektorom

$$P = [x(u)\ y(u)]$$

Slično tačka na krivoj u prostoru se predstavlja kao vektor

$$P = [x(u)\ y(u)\ z(u)]$$

a tačka koja pripada ravni predstavlja se pomocu vektora

$$P = [x(u\ v)\ y(u\ v)\ z(u\ v)]$$

Primena parametarskih jednačina ne samo da alimniše ranije pomenute probleme već predstavlja i najpogodniji oblik za prikazivanje na grafičkim perifelalima (terminalima i ploteima)

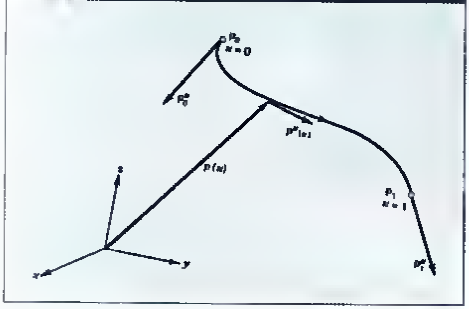
Ako posmatramo tačku kao osnovni element parametarske geometrije onda je moze mo definisati pomocu uredenog skupa realnih brojeva koji obično predstavljaju koordinate ta tacke u prostoru Matematička funkcija koja generiše skup tacaka koja definišu krivu ravan ili druge geometrijske elemente su parametarske jednacine Medutim polpuno je jasno da parametar u ne može uzimati rednosti od -0 do +0 dakle moramo izabrati interval definisanosti parametra u od znacaja za nas problem Iz praktičnih razloga pogodno je vrednosti parametra u normalizovati to jest ograničiti domen njegove delimisanosti na interval [0 1] kako je to prikazano na slici 1

Dakle osnovni razlozi zbog kojih se primenjuje parametarski prstup u geomatrijskom modeliranju su sledeći

1 Parametarske jednačine pruzaju više slepeni slobode u modeliranju nego neparametarske jednacine To se lako može videti na primer i iz same definicije eksplicitne forme za krivu trećeg reda

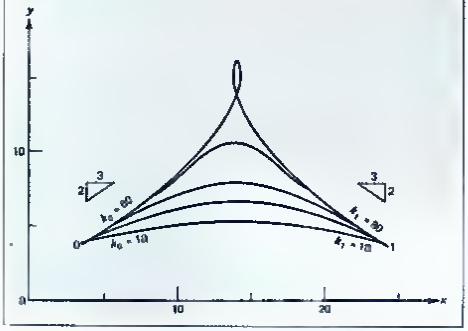
$$y = ax^3 + bx^2 + cx + d$$

Slika 1.



Normalizacija vrednosti parametara (ograničavanje definisanosti na intervai [0,1]

Slika 2



Promenom intenziteta tangentnih vektora, bez menjanja jediničnih tangentnih vektora, mogu se dobiti različiti oblici krive

Četiri koeficijenta **a b c** i **d** u ovoj jednačini određuju oblik krive dok u parametarskom obliku iste jednačine

$$x = au^3 + bu^2 + cu + d$$
$$y = eu^3 + fu^2 + gu + h$$

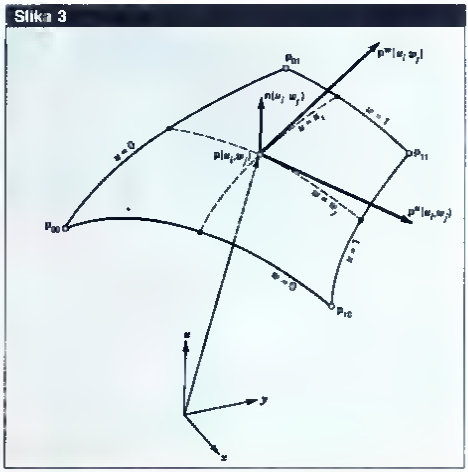
postoji osam koeficijenata kojim je moguce moguće menjati oblik krive Ovo znači da je u poslednjem slučaju mogućnost variiacije oblika mnogo veća

2 Uobičajeni postupci transformacije geometrijskih elemenata kao sto su translacija rotacija i skaliranje mogu se direktno vrsiti nad parametarskim jednačinama

3 Primenom parametarskih jednačina eliminiše se problem neodredenosti za lačke u kojima je nagib krive 90 stepeni

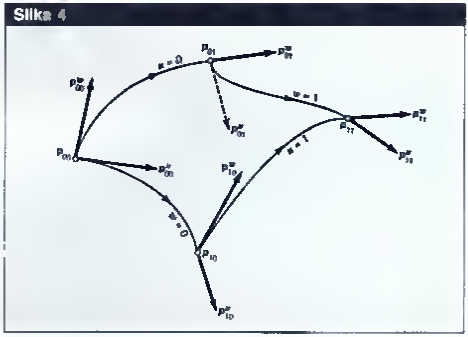
4 Parametarske jednačine omogućavaju definisanje krivih i povrsi u visedimenzionalnom prostoru bez promene njihovih geometrijskih karakteristika Mogucnost potpunog razdvajanja zavisnih i nezavisnih promenljivih u parametarskim jednacinama dozvoljava da se sa geometrijskim elementima operise u njihovoj geometrijskoj formi Konšlenjem određenih matematičkih meloda primenjenih na geometrijske forme (*blending functions*) pruža se mogucnost odlje kontrole oblika nego u klasicnim algebarskim formama

Slika 3



Za definiciju segmenta povrsi neophodno je definisati četiri granične krive i odgovarajuca temena koja se naiaze na presecima ovih krivih

Slika 4



Oznake za predstavljanje segmenta povrsi

5 Prilikom parametarskog definisanja geometrijskih formi nije potrebno voditi racuna o dimenzijama objekta jer operišemo sa normalizovanim parametarskim promenljivim

6 Geometrijske elemente opisane pomoću parametarskih jednacina vrlo je pogodno predstaviti u vektorskoj i matricnoj formi što je izuzetno pogodno za računarsku obradu

PARAMETARSKO DEFINISANJE KRIVIH LINIJA

Algebarski oblik parametarskog kubnog segmenta krive linije opisuje se pomoću sledeća tri polinoma

$$x(u) = a_{3x}u^3 + a_{2x}u^2 + a_{1x}u + a_{0x}$$
$$y(u) = a_{3y}u^3 + a_{2y}u^2 + a_{1y}u + a_{0y}$$
$$z(u) = a_{3z}u^3 + a_{2z}u^2 + a_{1z}u + a_{0z}$$

U gornjim polinomima vrednosi parametra u kao nazavisno promenljive variiable ograničena je u domenu [0 1] Iako da je segment krive ograničen sa dve definisane krajnje lačke za $u = 0$ i $u = 1$ Ovih dvanaest koeficijenata nazivaju se **algebarskim koeficijentima** a koji definišu oblik i poziciju krive u prostoru Dve krive istog oblika imaju različite algebarske koeficijenle i zauzimaju različite pozicije u prostoru Uobicajeni kondenzovani Io jesi vektorski način predstavljanja algebarske forme parametarske jednačine je

$$p(u) = a_3u^3 + a_2u^2 + a_1u + a_0$$

Vektor p(u) predstavlja bilo koju lačku na segmentu parametarske krive a a_3 a_2 a_1 i a_0 su veklorski ekvivalenti za algebarske koeficijenle Prvi izvodi u pojedinim lačkama krive (slika 1) definisani su kao

$$p^u(u) = dp(u)/du$$

a komponente ovog vektora kao

$$x^u = dx(u)/du$$
$$y^u = dy(u)/du$$
$$z^u = dz(u)/du$$

Veklori p(0) $p^u(0)$ p(1) i $p^u(1)$ predstavljaju granične uslove

Gore definisani algebarski koeficijenti nisu uvek pogodni za kontrolu oblika i prikazivanje parametarskih krivih. Za to je mnogo pogodniji geometrijski oblik parametarske krive koji omogućava upotrebu geometrijskih parametara kao što su granične tačke na segmentu, nagibi tangenta, torzija i zakrivljenost krive. Ako iskoristimo granične tačke na segmentu p(0) i p(1) i svakako izbegnemo objašnjavanja matematičkih „sallo mortalla" operacija, dobijamo takozvanu **Hermiteovu krivu** (francuski matematičar iz XIX veka koji se bavio problemom kubnih polinoma), u obliku

$$p(u) = (2u^3-3u^2 + 1)p(0) + (-2u^3+3u^2)p(1) + (u^3 2u^2 + u)p'(0) + (u^3- u^2)p (1)$$

Hermiteova jednačina može se predstaviti u geometrijskom obliku

$$p = F1p_0 + F2p_1 + F3p_0^u + F4p_1^u$$

$$F1(u) = 2u^3 - 3u + 1$$

$$F2(u) = -2u^3 + 3u^2$$

$$F3(u) = u^3 - 2u^2 + u$$

$$F4(u) = u^3 - u^2$$

gde F1, F2, F3 i F4 predstavljaju takozvane **funkcije mešanja** (*blending functions*) a p_0, p_1, p_0^u i p_1^u geometrijske koeficijenta.

Navedenu jednačinu možemo napisati u kondenzovanoj formi kao

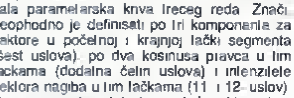
$$p(u) = U \times W \times B$$

gde je $U = [u^3\ u^2\ u\ 1]$, a **M** je takozvana univerzalna transformaciona matrica koja je jednaka

$$\mathbf{M} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

a **B** predstavlja matricu geometrijskih koeficijenata ili granične uslove i iznosi $\mathbf{B} = [p_0\ p_1\ p_0^u\ p_1^u]^T$. Ovo znači da matrični oblik parametarske jednačine prave ima oblik

formula



Ako sa $A = [a_3\ a_2\ a_1\ a_0]^T$ označimo vektore čiji su članovi jednaki algebarskim koeficijentima onda se jednačina prave u parametarskom obliku može napisati u formi

$$p = U \times A$$

Bilo je učiti da su matrice **U**, **F**, **M** identične za sve parametarske krive, dok se matrice **A** i **B** razlikuju od kriva do krive.

Analizom gornjih jednačina vidimo da je neophodno odrediti 12 uslova, da bi se definisala parametarska kriva trećeg reda. Znači, neophodno je definisati po tri komponente za vektore u početnoj i krajnjoj tački segmenta (šest uslova), po dva kosinusa pravca u tim tačkama (dodatna četiri uslova) i intenzitete vektora nagiba u tim tačkama (11 i 12. uslov). Promenom intenzitata tangentnih vektora, bez menjanja jedinićnih tangentnih vektora, mogu se dobiti različiti oblici krive (slika 2).

Kriva linija dobijena na ovaj način, kao što se vidi, zahteva definisanje tangentnih vektora u krajnjim tačkama svakog segmenta. To jest da bi se dobila sastavljena kriva neprekidnosti C neophodno je da su jedinični tangentni vektori na spoju dva segmenta jednaki. Na ovaj način dobijena kriva praktično interpolira zadane tačke. Osnovna slabost ovakvog pristupa je u teškoći prilikom određivanja tangentnih vektora, mada je direktno navođenje tangentnih vektora moguće izbeći primenom specijalnih algoritama (npr. Coons-Ahujin-ovog).

Gornja konstatacija je izuzetno bitna jer novije metode (koje će biti objašnjene sledećem članku) imaju potpuno drugačiji postup u definisanju krivih linija.

PARAMETARSKO DEFINISANJE POVRŠI

Ono što u definisanju krivih linija predstavlja segment krive, to u definisanju površi predstavlja „patch", odnosno segment površi (slika 3). Navedeni segment definiše se kao skup tačaka ograničen krivim linijama i definisan dvoparametarskim jednačinama određenim funkcijama

$$x = x(u,w),\quad y = y(u,w),\quad z = z(u,w)$$

Kao što se vidi sa slike 3, za definiciju segmenta površi neophodno je definisati četiri granične krive i odgovarajuća temena koja se nalaze na presecima ovih krivih.

U algebarskoj formi segment površi se predstavlja kao

$$p(u,w) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} u^i w^j$$

gde $u, w \in [0\ 1]$

Ovaj segment se često naziva i bikubni segment jer se oba parametra (u,w) javljaju na trećem stepenu dok a predstavlja algebarske koeficijente površi. U matričnoj formi gornju jednačinu možemo predstaviti kao

$$p = U\ A\ W^T$$

gde je

$$U = [u^3\ u^2\ u\ 1]$$

i

$$W = [w^3\ w^2\ w\ 1]$$

$$A = \begin{bmatrix} a_{33} & a_{32} & a_{31} & a_{30} \\ a_{23} & a_{22} & a_{21} & a_{20} \\ a_{13} & a_{12} & a_{11} & a_{10} \\ a_{03} & a_{02} & a_{01} & a_{00} \end{bmatrix}$$

Na osnovu matrice **A** vidi se da u algebarskoj formi bikubnog segmenta površi postoji šesnaest koeficijenata od kojih svak ima po tri komponente što znači da ovakav segment ima čelrdeset osam stepeni slobode. Kao što je već rečeno, u definisanju krivih linija algebarski koeficijenti definišu oblik segmenta površi i njegov položaj u prostoru. Međutim mnogo pogodniji pristup u kontrolisanju oblika segmenata površi pruža njihovo predstavljanje preko geometrijskih koeficijenata. Da bi se predstavio segment površi upotrebljavaju se oznake kao na slici 4.

Korišćenjem nomenklatura sa prethodne slike granične krive možemo

$$P_{u0} = F\ [\ P_{00}\ P_{10}\ P_{00}^u\ P_{10}^u\ ]$$

$$P_{u1} = F\ [\ P_{01}\ P_{11}\ P_{01}^u\ P_{11}^u\ ]$$

$$P_{0w} = F\ [\ P_{00}\ P_{01}\ P_{00}^w\ P_{01}^w\ ]$$

$$P_{1w} = F\ [\ P_{10}\ P_{11}\ P_{10}^w\ P_{11}^w\ ]$$

Definicijom graničnih krivih određeno je dvanaest od šesnaest vektora neophodnih za određivanje čelrdesel osam geometrijskih koeficijenata. Ie su potrebna četiri dodatna vektora. Zbog ovoga se uvode takozvani vektori uvijanja površi u temenima segmenta površi, a koji se definišu kao

$$P_{00}^{uw} = \frac{\partial^2 p(u,w)}{\partial u \partial w}$$

za u = 0, w = 0

$$P_{10}^{uw} = \frac{\partial^2 p(u,w)}{\partial u \partial w}$$

za u = 1, w = 0

$$P_{01}^{uw} = \frac{\partial^2 p(u,w)}{\partial u \partial w}$$

za u = 0, w = 1

$$P_{11}^{uw} = \frac{\partial^2 (u,w)}{\partial u \partial w}$$

za u = 1, w = 1

Sa definicijom vektora uvijanja zatvoren je sistem (definisano je svih šesnaest vektora), pa je sada moguće predstaviti bikubni segment površi u matričnoj formi kao

$$p = U\ M\ B\ M^T\ W^T$$

Matrice **U**, **M** i **V** su poznate iz definicije segmenata kriva pa je neophodno samo dati definiciju matrice geometrijskih koeficijenata koji određuju bikubni segment površi

$$B = \begin{bmatrix} P_{00} & P_{01} & P_{00}^w & P_{01}^w \\ P_{10} & P_{11} & P_{10}^w & P_{11}^w \\ P_{00}^u & P_{01}^u & P_{00}^{uw} & P_{01}^{uw} \\ P_{10}^u & P_{11}^u & P_{11}^{uw} & P_{11}^{uw} \end{bmatrix}$$

Prva kolona u matrici B određuje graničnu krivu p_{u0}, druga kolona određuje drugu graničnu krivu p_{u1}, treća kolona određuje pomoćnu krivu koja se označava sa p^w_{u0}, a četvrta pomoćnu krivu p^w_{u1}. Prva vrsta u matrici geometrijskih koeficijenata B predstavlja graničnu krivu p_{0w}, druga graničnu krivu p_{1w}, a treća i četvrta pomoćne krive p^w_{0w} i p^w_{1w}.

Podelom matrice B na četiri kvadranta vidi se da prvi kvadrant (gornji desni) definiše temena segmenta, gornji desni tangentne vektore u odnosu na parametar w, donji levi tangentne vektore u odnosu na parametar u, a donji desni vektore uvijanja segmenta površi.

Gora navedeni matematički opis segmenta površi predstavlja tzv. Hermiteov bikubni površinski segment i nema neku veću primenu jer je neophodno zadavati izvode a nema mogućnost osobitog lokalnog nadzora. Ova vrsta površi interesantna je sa stanovišta analitičke upotrebe jer ona interpolira sve zadane tačke.

Ako dva površinska segmenta želimo spojiti u jedan sa neprekidnošću C^1, odgovarajuće granične krive linije moraju biti jednake a pomoćne krive proporcionalne.

MATEMATIČKI PRESING

Za poverovati je da je odrađen broj čitalaca *Računara* teško podneo ovaj matematički pristing. Međutim smatramo da je bio ipak potreban najmanje iz dva razloga. Prvo izložene su matematičke osnove na kojima se baziraju sva ostale metode definisanja krivih i površi u procesu geometriskog modeliranja objekata. Drugo metode koje su predstavljene u ovom članku daliraju iz šezdesetih godina i krairali su ih vrhunski matematičari na MIT-u (Coons, Ferguson i drugi) što govori da su bile vrlo nepnmerene svakodnevnoj interaktivnoj upotrebi. Inženjeri su u toku projektnog postupka morali da barataju formulama a promena oblika krive ili površi zahtevala je dosta sa današnjeg aspekta nepotrebnog truda. Sve ovo govori da je bilo neophodno stvoriti nov pristup u samom definisanju problema, o čemu će biti više reči u sledećem članku iz ove serije.

LOKALIZOVANI MEKINTOŠ

Iako se „Računari" isključivo bave IBM PC standardom, ne bi bilo loše da, s vremena na vreme, zbog opšte informatičke kulture, bacimo pogled i na to šta radi „suprotna strana". Inovirani operativni sistem za jedan od najslavnijih računara svih vremena je, svakako, vredan naše panje.

Robert Slevečki

Iako je Apple već više od dvije godine pompozno najavljivao izlazak nove verzije sistema za svoja Macintosh računala, Sistem 7.0 se na američkom tržištu pojavio tek početkom ovog ljeta. Zbog lokalizacije izvan američki konsnici pa tako i mi morali su pričekati još koji mjesec.

Na početku je važno napomenuti da sistem 7.0 nije namijenjen samo za rad sa najjačim Macintosh modelima kao što je OS/2 za jake Intel-bazirane sisteme. Istina, sedmica je nešto zahljevnija od njezinih prethodnika, ali ona će unatoč tome raditi na skučenom Classicu sa hard diskom i 2 megabajta RAM-a, kao i na maksimalno frizriranom Macintoshu IIfx. Mislim da je nepotrebno napominjati da je novi sistem u potpunosti kompatibilan sa ranijim verzijama iz čega odmah proizlazi da sve „stare" aplikacije možete koristiti i u novom okruženju.

Međutim, način na koji Apple namjerava distribuirati Sistem 7.0 znatno se razlikuje od načina na koji se to radilo sa ranijim modelima. Kada za koji ljedan kod svojeg dilera (a može već i u vrijeme dok čitate ovaj članak) kupite sedmicu, bit će to isključivo u vidu lokalizirane verzije. Drugim riječima, jedina verzija do koje će ići legalno u Jugoslaviji bit će ona koje će umjesto običajnog „FILE EDIT " sadržavati „DATOTEKA UREDNIK " Nadalje lokalizacija, odnosno prijevod, neće se odnositi samo na software već i na priručnike. Iako ovaj Appleov potez ima i dobrih strana (lakše korištenje, obzirom da konsnik ne mora poznavati engleski jezik), on ima i svojih nedostataka. Kao prvi mogu navesti činjenicu da ljude koji su sa naviknuli na rad sa engleskim izrazima nerviraju naši nazivi poput „DATOTEKA" „UREDNIK" ili npr. „ZEMLJOVID" Istina, nad ovim se problemom može i zažmiriti, ali pravi problem predstavljaju izuzetno popularne aplikacije (Poput *Page Makera* ili *QuarkXPressa*) koja ili uopće ne rade sa postojećim lokaliziranim verzijama, ili rad sa njima predstavlja odredene probleme. Dok na taj način *Page Maker* pod YU-sistemom 6.0X neće numerirati stranice, *QuarkXPress* je golovo nemoguće pokrenuti. Možemo se samo nadati da će ljudi koji rade na lokalizaciji sedmice riješiti ove probleme. Iako ovo pitanje svekako ostaje: novi sistem posjeduje enormnu količinu novih funkcija koje će poknti navedene nedostatke.

NOVI FINDER

Promjene koju ćete svekako uočiti nakon nekoliko sekundi rada sa novim sistemom je dostalno izmijenjen dizajn. Kao prvo, Finder više ne postoji. U stvari on i dalje postoji, ali ne u smislu Finder/MultiFinder. Pod Sistemom 7.0 Finder je MultiFinder. Ovaj novi Finder je pun poboljšenja koja očito dokazuju da Apple sluša zahljeve tržišta. Možda je najvažnije napomenuti da on sada ima znatno poboljšane funkcije. Multitaskinga. Iako da sada možete bez problema izvršavati i više zadataka odjednom. Ikone i prozori sada imaju trodimenzionalni izgled, sa više detalja i boje. (U svakom slučaju, da bi u potpunosti uživali u ovim različitim kama potrebno je raspolagati sa najmanje 16 boja ili sivih skala.) Pa ipak, općenito gledano, Apple je zadržao sada već klasičan desktop po kojem uostalom i prepoznajemo Macintosha, a čije su osnovne odlike jednostavnost i nenamelljivost.

Iako je meni „COLOR" sada zamijenjen sa „LABEL" cjelokupni izgled osnovnog menija ostao je nepromijenjen. Ukoliko međutim otvorite neki od menija odmah ćete uočiti znatne izmjene. Prvo, kraj svakog pojedinog naziva u meniju nalaze se sialke male ikonice. Drugo: Apple meni nije više samo za desk acesorije (DA), u njega sada možete povući (nema više Font/DA Mover-a!) bilo koju aplikaciju ili dokument koristeći Menu Iolder koji se sada nalazi u sistemskom folderu. Ove su promjene trenutne. Iz sada više nemorate restartati vaš Macintosh kako bi konsnili koristili novoinstalirane fontove ili druge desk acesorije.

Nestao je također i svugdje prisulan Control Panel. Umjesto njega Apple je ugradio meni opciju nazvanu Control Panels, sada smještenu kao i Menu Folder u Sistemsko Folderu (pogledajte sliku). Ovu je melalom Apple izabrao i za sve ostale Sistemske Modifikatore – sve funkcije ove vrste sada je smjestio u zasebne foldere: unutar Sistemskog Foldera. Ovom organizacijom Apple je omogućio znatnu fleksibilnost te mogućnosi velikih modifikacija unutar samog sistema. Promjene koje ste prije mogli izvršiti samo uz pomoć zasebnih sistemskih alata ili aplikacija sada možete napraviti koristeći funkcije iz Sistemskog Foldera. Na taj način možete promijeniti sistemski font. (Do sada vec istrošeni Chicago i Geneva) mijenjati boju prozora, njihovih okvira te ikona i dr.

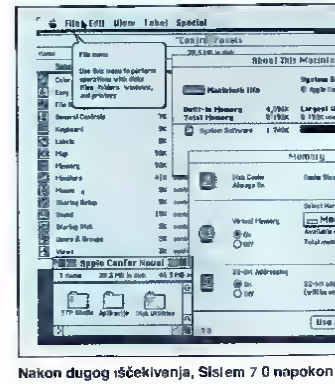
Ono što je posebno zanimljivo je i činjenica da je u Sistem 7.0 sada ugrađena i kompletna podrška za rad u mreži. Iako da sada nema potrebe za korištenjem aplikacija kao što je npr. AppleShare. Dapače, nove funkcije unutar sedmice ne zahljevaju čak niti dedlinarje servera. Dovoljno je samo odabrati folder koji želite sa nekime razmijeniti te zatim odabrati opciju „Sharing " koja se sada nalazi u meniju FILE. Tu je još i Data Acces Manager pomoću kojeg sade možete na lak način razmjenjivati aplikacije sa drugim strojevima u mreži. Cijeli ovaj mrežni model sadrži i sistem zaštite tako da sve svoje vitalne datoteke možete zaštititi od nezeljenog skidanja.

OSTALI NOVITETI

Osleta poboljšanja unutar Findera uključuju i izuzetno brzu i snažnu FIND naredbu (Koja vam sade ne daje informacije o tome gdje se nalazi tražene datoteke, već vas i odmah odvodi do nje.) Posebno je zanimljiva mogucnosl tzv. Fantomskog kopiranja, odnosno kreiranja malih. Alias datoteka dvojnika od 2K. Na taj način možete npr. na diskeli kreirati dvojnike svih aplikacija koje se nalaze na hard disku i kasnije pozivati le aplikacije preko njihovih dvojnika. Tu su dakako i nove memorijske mogućnosti.

Za razliku od prethodnih sistema, koji ne podržavaju virtualnu memoriju i ograničavaju vas na tzički adresni prostor od 8Mb. Sistem 7.0 posjeduje jednostavnu implementaciju za rad sa virtualnom memorijom koju je moguce adresirati do 1 gigabajta, kao i podržavanje realne memorije do 32 Mb. Međutim, rad sa virtualnom memorijom moguc je samo na 68030-sistemima kao i onima sa 68020 procesorom uz eksternu MMU (Memory Managament Unit) Iako ce virtualna memorija alimirali mnoge probleme (posebno pri radu sa npr. 24-bitnim slikama) nemojte od nje očekivali čuda. Svaka usporedba u brzini rada između virtualnel realne memorije mogla bi vas košlati živaca.

Windowsi su se čak i u najrenjim verzijama mogli pohvaliti sa dinamičkim povezivanjem aplikacija (DDE – Dynamic Data Exchange). Ovaj skup funkcija istina pod drugačijim imenom (IAC – InterApplication Communications) sada posjeduje i Sistem 7. IAC će naime omogucavati interaktivnu kumunikaciju između aplikacija po principu izdavač-Prelplatnik. Uzmimo za primjer da radile sa dvije aplikacije, nekim programom za crtanje, npr. X i programom za stranicni prelom, npr. Y. Oba su programa istovremeno otvorena. Ie u programu X selektirale vašu sliku i odaberale opciju „PUBLISH" zatim prelazile u progam Y i aktivirate „SUBSCRIBE"



Nakon dugog iščekivanja, Sistem 7.0 napokon se probija na tržištu

Nastavlja se na strani 77

UZBRANO S DRVETA

U vreme kada i najvarniji mornari veštačke inteligencije polako nspuštaju svoj lep i voljeni ali ne odveć koristan (bar za putnike) brod, izgleda da se posadi pridružio izuzetno uticajan kapatan — Danny Hillis koji je prvi uspeo da uzgoji računarski program simuiacijom procesa biološke evolucije.

Saša R. Jović

Ako se nekoma Ima učini poznato. iako nikade nija pogledao u knjige i časopise sa tematkom veštačka inteligencija to nija slučajno Hillis je osnivač i glavni naučnik (što je vrlo česta titula za onog ko je prvi lansirao ideju iza produkta firme u manjim prodorim kompanijama u Severnoj Amerci) Thinking Machines Inc Naravno. on je i dizejner prve Connection Machine superkompjutera sa 64 K procesora Ovog puta on je rešio da krene lamo odakle mnogi beže Prvi korak je bio kreiranje močnog i efikasnog programa za sortiranje koji je automatski generisan („uzgojen") uz pomoć simulacije procesa biološke evolucije. Odmah da kažemo da se njegov pristup značajno razlikuje od mnogobrojnih varzija simulirane evolucije i genetskih algoritama koji su već duže vreme popularni u raznim optimizacionim problemima upravo po tome što je Hillisov cilj ne rešavanje određenog kombinatonalnog zadatka, već automatsko generisanja algoritma za rešavanje takvih problema U određenom smislu la nova oblast mogla bi se nazvali metaalgoritmika

Naravno sama ideja da pnroda krije tajnu veštačke inteligencija i da tieba pažljivo raditi na analogijama koje ona pruža nije nimalo nova Najrasprostranjenij i trenutno najpopularniji primei su neuralne mreže Međutim u Hillisovom pristupu ima mnogo novih i interesantnih elemenala Pokušavajući da kreira ljudski nivo inteligencije na mašini on je održao nekoliko predavanja na fakultetima na Istočnoj obali Amerike. (Glasine kažu da se nje usudio da ode na Zapadnu obalu jer tamo te automatski posmatraju sa preziorm ako pokušaš da im ponovo prodaješ veštačku inteligenciju I ne samo sa prezirom već ponekad i sa ljutnjom. jer nisu ralki oni koji su izgubili pare finansirajući neuspešne pokušaje u doličnoj oblasti) Naslov predavanja je više psihološko-bibljiski nego računarski „Inteligencija kao ponašanje koje se pojavljuje ili pesme Edena"

Konsteći novi programski jezik za Connection Machine koji znalno olakšava eksperimentlisanje algoritmima. Hillis razvija modele veštačke evolucije, nadajući se da će konačno otkriti tajne koje decenijama izmiču istraživači ma na MITu Interesantno je da je i on počeo svoju naučnu karijeru na MITu doktoratom čija je teza bila Connection Machine

1. Mistika

Govoreći pred dupke punim dvoranama, Hillis je rekao „Danas su istraživači u veštačkoj inteligenci podeljeni u dva tabora one koji koriste simboličke metode i gde je sva poznato i tačno određeno i one koji koriste neuralne mreže gde je naglasak priznavali mi to ili ne na mističnom elementu U opštam slučaju mi ne znamo kako neuralne mreže rešava problem i to je jedan od glavnih razloge za njihovu nedavnu popularnost S druge tačke gledišta mi verujemo da je mozak mašina jer to nas uče modarna nauke i tehnologija I opet u isto vreme mi ne znamo principa rada te mašine i verujemo u dušu " (Obično se u računarskim časopisima kada nešto nije opšte znanje, a nije lako za razumevanje pri prvom čitanju referše na metemetičke ili tehnološke reference. U ovom slučaju, ako imala problema sa praćenjem osnovnih ideja, možde je najbolje da konsultujete teološke, ideološke ili bar psihološke knjige)

Ali Hillis veruje da su ova dva pristupa znalno slična nego što njihovi proponenti žele da priznaju Inteligentni sistemi su odlični kandidati za platformu za biološki inspirisane procese koji mogu da transcedentiraju naša poimanja kako to radi Inteligencija

Zalo Hillis naslavlja „Osnovne i ključna ideja za dalji napredak vištačka inteligencije je sludiranje i razumevanje pojavljivanja ponašanja. U fizičkim i biološkim sistamima, mi nalazimo široku lepezu kompleksnog ponašanja sistema. Kada se u takvim sistemima primeni mali broj jednostavnih pravila može se dobiti vrlo kompleksno ponašanja Na primer, fraktalni oblici pahulja i oblici lisća prilikom kretanje čamca su posledice vrlo jednostavnih pravila primenjenih na molekule vode " Jezik jedno od osnovnih oruda ljudske inteligencija je isto tako primei ponašanja koja se polako pojavljuje Hillis kaže da se jezik najverovatnije razvio tako što se nekoliko jednostavnih pravila za selekciju razumljivih od nerazumljivih glasova primenjivelo tokom generacija naših predaka

Citajmo dalje Hillise. Zamisite vreme — negde pre jedno milion godina — kada su majmuni počeli da hodaju uspravno Oni su imali inteligenciju na nivou današnjih šimpanza Zatim su počeli da proizvode razno-razna zvukova Ti zvuci — pesme Edena — nisu imali nikakvo ili skoro nikakvo inteligentno značenje Majmun definitivno nije imao namenu da prenese bilo kakvo sofisticirano značenje pomoću zvukova koja je puštao (Za ona kojima se ova teorija ne svida ili jednostavno misle da je pogrešna molimo da kontaktiraju samog Hillisa ne uredništvo „Racunara") Ali kroz tendenciju da imitiraju umilne zvukova i izbegavaju i zaboravljaju namelodične majmuni su neholiko krallali rastući medij koji je omogućio kasnu evoluciju današnjih kompleksnih jezika Zvuci koji su bili korisnji u zajednici majmuna bili su najčešće ponovo korišćeni i formirali su bazu za razvoj inteligencije Ovaj scenario implicira da se jezik nije razvio tako što su već inteligentna bića rešila da govore već da se razvijao zajedno sa ljudskim mozgom "

Na osnovu ove i slično bazirarih interpretacija o razvoju prirode i prirodne inteligencije (bilo kod ljudi ili životinja), Hillis zaključuje „Ako bismo kreirali okolinu u kojoj bi relativno neinteligentna racunarska tehnologija bila inficirana našom ljudskom inteligencijom, mogli bismo konačno da ostvarimo dobitnu kombinaciju veštačke inteligencije Uzgajajući umesto konstruišući veštačku inteligenciju, istraživači bi bili sposobni da naprave novi nivo inteligencije i bez razumevanja kompleksnog simboličnog računa i bez razumevanja kako biološka inteligencija radi"

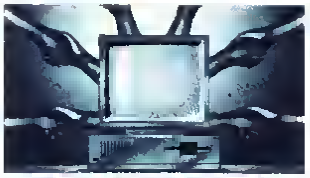
2. E, a sad nešto konkretno

Da bi pokazao da njegova teorija nije samo prazno mlađenje filozofske slame Hillis je „uzgajio" algoritam za sortiranja uz pomoć veštačke evolucije Hillis je prvo počeo sludirajući ponašanje raznih biološko inspirisanih algoritama ne Connection Machine a zatim je dodao u ceo proces grupu matematičkih pojmova koji su dovoljni da bi se definisao sortirajuci algoritam [iako je ovde verovatno to ključni elemenal kako prepoznali prave matematičke pojmove koji su potrebni i dovoljni da bi uzgajio određen algoritam Hillis nija dalje alaborisao ovaj korak Naravno, iako se može desiti da je ovaj korak u generalnom slučaju isto toliko lažak ili leži nego garanisanja celog algoritma eli nemojmo pakosnim (iako možda lačnim) primedbama da kvalimo lepu ideju)

Sledeći korak je bio primena algoritma koji se spontano definisao uz pomoć slučajne selekcije podskupa matematičkih pojmova, na razna nizova brojeva Ako algoritam ne bi bio uspešan pri sortiranju, iz skupa matematičkih pojmove bi bio uzet novi matamatički pojam za predafinisanje algoritma Među već konščenim matematičkim pojmovima bi sa izbacivao onaj koji, kada se ne primeni najviše doprinosi boljem rezultatu. Ova procedura se ponavlja sve dok se nije dobio biz i pouzdan algoritam za sortiranje Procedura je analogna se seksualnom reprodukcijom pri čemu je individua algoritam a virusi koji napadaju i lestiraju sposobnosl individue nizovi lestirajućih brojava Konačno se polpuna analogija uspostavlja kada se izuzelni kvalitet koji omogućavaju savladivanje problema koje uzrokuje okolina izedače as matematičkim pravilima

Seksualna reprodukcija ubrzava stvaranje generacije sposobnih potomake ne taj način što omogućeva rodtelima da konceotnšu svoju pažnju i prenesu znanje na najsposobnije i najperspektivnije potomke kaže Hillis (Ovo sve mozda pomalo podseca na lazne rasističke i nacionalističke prislupe ali podselimo da se ovde radi o algoritmima ne ljudima i da cilj i kriterijumi pri razvoju algoritama na moja biti i najčešće nija isti kao pri razvoju čovečanstva) Interesantno je pri tom napomenuti dva zaključka koja je Hillise našao prilikom svog istraživanja Raznoliki i teški virusi (nizovi brojeve za sortiranja) znatno uzbrzavaju razvoj algoritama Bez seksualne isprodukcije nje moguće postići kvalitelen algoritam — jednostavno evolucija se zaustavlja i polomci velno reprodukuju svoje rodtelja suviše lano pre nego što se postigne sposobnosi sortiranja laških nizove brojeva

Da Danny Hillis nije sve suviše pojednostavio i prenebregao važne, odlučujuce dalalje govori nastavak njegovog izlaganja Naime sve ovde do sada izloženo izgleda kad tipičan „greedy" (pohlepan t lakav da sa uvek uzima pravac najvećeg uspeha) princip za kreiranja algoritama Tekav pristup se po pravilu pri primeni kod skoio svih problema od praktičnog interesa zaustavlja u prvom lokalnom minimumu Naravno lokalni minimum je skoio uvek dalako infenomiji u odnosu na globalni mini-



mum. Računerdžije su davno zaključile: ako se pri ovom prslupu (sekvencijalno poboljšavanje) ne uvede mehanizam koji omogućava da se povremeno prihvate i slabija rešenja da bi se izbegli lokalni minimumi, rezultati su za sve sem najjednostavnijih problema očajni. (Interesantno je da su za najjednostavnije probleme ovi algoritmi vrlo često najbolji i najbrži od svih algoritama.)

Jednom kad je prirodna selekcija kreirala najboljeg pojedinca, morao je da se uspostavi princip koji podržava raznovrsnost. (Jednostavno, bez raznovrsnosti postižu se kratkoročno odlični rezultati, ali dugoročno rezultat je po pravilu poražavajući i očajan — ovoga puta izgleda da poredenje algoritama sa ljudskim društvima možda i nije tako besmisleno). Znači, rani kvaliteti koji su jednom odbačeni, jer su bili nepotrebni za savladivanje okoline (niza brojeva) u jednom trenutku, moraju se s vremena na vreme opet uzeti u razmatranje. Jednostavno, moraju postojati i drugi faktori pri evoluciji sem prirodne selekcije.

3. Mali zeleni (van) zemaljci

Zato Hilis kaže „Jednostavni jednoćelijski zeleni organizmi imaju optimalne šanse za opstanak na današnjoj planeti Zemlji. Kad bi prirodna selekcija bila jedini mehanizam evolucije, tada bi ovi zeleni organizmi bili jedini živi organizam, i Zemlja bi bila uniformno pokrivena debelim zelenim pokrivačem. Međutim, očigledno to nije slučaj, i zato moramo potražiti koji ostali principi vladaju u prirodi sem prirodne selekcije"

Naravno, naći ove ostale faktora nije ni lako ni brzo ni jednostavno. Ali Danny Hillis smatra da je ova potraga i te kako vredna napore. Njegov moto je jednostavan. „Ne mislim da treba izbegavati neka pitanja samo zato što ne znamo odgovor na njih." Pri ovoj potrazi Hillis ima i nekoliko relativno novih ideja, koje su potencijalno vrlo korisne za razvoj algoritama i veštačke inteligencije u budućnosti. Jedna od takvih je da ako želimo što bržu evoluciju nije dovoljno samo evoluirati organizam koji se razvija, nego i viruse koji ga napadaju, tako da virusi postaju sve sofisticiraniji i teži za rešavanje. Tako će oni uvek nelaziti novi način da napadnu organizam, što će motivirati organizam da se još usavrši.

4. Jednostavna matematička forma

Pri testiranju svojih ideja, Hillis je namerno konstio vrlo jednostavne matematičke forme što mu je omogućavalo više slobode pri delinisanju genetske prilagodenosti na okolinu i faktora prilagodenosti. Virusi su takode bilo vrlo jednostavni. Konstećl Connection Machine on je na taj način mogao da prode kroz hiljade generacija u sekundi. Rezultati su prikazivani na kolor monitoru, tako da je Hillis mogao da celu evoluciju posmatra u realnom vremenu i da razvije intuitivni osecaj za razne evolucione modele. Sledeći korak je bio da se posle razmatranja vrlo jednostavnih modela i sticanja znanja na njima pokuša trelranje pravog teškog problema iz svakodnevnog života. Na taj način bi se postigao inicijalni cilj.

Hillis je izebrao sortiranje kao test za dokaz koncepta, zato što se ovaj problem često koristi kao test primer pri dizajniranju i procenjivanju paralelnih računara. Naravno, zadatak je bio da se uzme proizvoljan niz brojeva i da se oni poredaju u rastućem (ili još preciznije u neopadajucem) poretku, u zavisnosti od njihove vrednosti. U poslednjih trideset godina algoritmi za sortiranje su studirani u celom svetu sa velikom pažnjom. Mnogi veliki računarski umovi su se okušali na ovom problemu. Ne taj način razmišljao je Hillis: moći ću da uporedim koliki je stvarno dostignut nivo veštačke inteligencije.

U ovom trenutku treba napomenuti da je arhitektura Connection Machine posebno dobro prilagodene za sortiranje brojeva. Možda je i to imalo uticaja na Hillisovu odluku. Takoda: izbor problema je možda vrlo nesrećan jer posluji opasnost — kao što se ispostavilo u EURISCO da je zapanjujuća matematička kreativnost programa posledica funkcija delinisanih u LIPSu a ne izvanredne genijalnosti Lenata i njegovog metoda — da se možda ispostavi da je arhitektura Connection Machine možda više zaslužna i odgovorna za uspeh u kreiranju brzog programa za sortiranje nego Hillis i njegova ideja. Ali i u tom slučaju će Hillis profitirati — selimo se da je Lenat, pre nego što je greška otkrivena, obezbedio sebi značajnu naučnu karijeru i lepo plaćenu poziciju.

No vratimo se samom metodu. Sortiranje je kao problem pogodno za test primer iz još nekoliko razloga. Među njima je svakako važan razlog da je relativno jednostavno generisati raznovrsne viruse. Isto tako, rezultat je lako proveriti: ako neki algoritam ne može da sortira neku niz brojeva, treba ga modifikovati. Koliko? Pa, verovatno u onoj meri u kojoj se javljaju greške u sortiranju. Isti tako, nizovi brojeva se mogu jednostavno podešavati tako da napadaju algoritme koji izgledaju obecavajuće a zapravo nemaju sposobnosti da stvarno sortiraju sve nizove brojeva. Hillis je primetio da je na kraju imao ne samo izvrstan algoritam za sortiranje nego i izvanredne testove koji su na već poznatim algoritmima uzimali najviše vremena za sortiranje. Napomenimo da je na nizovima slučajnih brojeva i posebno na test sekvencama finalni algoritam bio skoro brz kao najbolji ljudski algoritmi.

Na kraju Hillis zakjučuje „Najinteresantnije stvari oko algoritma su da je ne znam kako algoritam radi. Ako se ispostavi da postoji neki novi niz brojeva za koji on radi, ja ne bih znao kako da likširam radi algoritma. Struktura algoritma izgleda vrlo neregularna, na prvi pogled, ali kada sam počeo da je pažljivo razmatram, otkrivao sam sve više i više pravilnosti. Vešlački uzgajen algoritam nije simetričan i elegantan kao čuveni Knuth Batcher algoritam (naravno, tako nismo to naglasili), zadatak je bio napisati (uzgojiti) algoritam za paralelnu mašinu, a to je znatno teže nego za jednoprocesorsku. To nismo posebno naglašavali, jer se valjda podrazumeva da danas niko više (ili ne bar autor računara sa hiljadama procesora) ne misli na jednoprocesorske mašine, ali on je brži ako se meri vreme na Connection Machine. Smeiram da mogu da izjavim da sam kreirao računarski sistem koji **transcendira** moju inteligenciju i da mogu da budem optimista da možemo kreirati novi nivo veštačke inteligencije i možda jednog dana i postici pravu ljudsku inteligenciju u mašini koju je napravio čovek."

Interesantno je da je prilikom Hillisovog govora postavljeno mnogo pitanja, ali nakon ovog zaključka jedini odgovor publike bio je dugotrajan pljesak.

PITALICA

Zamislite da se Cray, Digital, Compaq i desetak start-upova iz silicijumske doline udruže sa tucetom velikih japanskih kompanija (NEC, Toshiba, Hitachi, Fujitsu, Matsushita, Mitshubishi, Sharp...) i napravе personalni kompjuter. Računar ima superskalarni superpajpfanirani procesor koji koristi Jozefsonov spoj na superprovodniku koji na sobnoj temperaturi radi na 500 GHz. Uz multidimenzionalni holografski RAM od 500 TB i par hiljada terabajta silicijumskog diska, ima rezoluciju 2048 X 2048. Uz to, kompajler globalno automatski optimizuje sve jezike – od kobola do lispa i prologa – i sve prebacuje za dokumentaciju u C++ i adu i, naravno, sve prirodne jezike. Korisnički interfejs koristi isključivo prepoznavanje govora, ali ima i desetak raskolpnih tastera raznih tipova koji preuzimaju interfejs po želji korisnika. Naravno, tu su i par opcionih miševa i elektronsko pero.

Ceo računar staje na dlan, ima satelitske i fiber priključke kapaciteta par TB u sekundi, desetak koprocesora za neuralne mreže, obradu signala, fuzzy logiku, trodimenzionalnu grafiku. Konsti isključivo sunčavu energiju, koju čuva u litijumsko-magnezijumskim betanjama kapaciteta par GWh. Potrošnja energije par milivata na sat. U 5000 TB HEMT ROMa ima smešteno Kongresnu biblioteku. Lenjinovu biblioteku i biblioteke svih gradova širom sveta. Naravno, tu je i cela produkcija Holivuda na disku, kao i ostala zabavna produkcija.

Naravno, zahvaljujući potpuno automatizovanoj proizvodnji, košta 50 dolara.

E sad, šta mislite koje bi bilo prvo pitanje koje bi računerdžije širom sveta postavile pre kupovine.

Pa da, naravno. „Da li je IBM kompatibilan?"

ONA TRAŽI MUŽA

Znam da je vrlo neubičajeno da računarski časopisi objavljuju male oglase lične prirode, ali moja bliska prijateljica Keli, velika zaljubljenica u računare i računarstvo me je zamolila da ako mogu protumim ovo uredniku. Kako je Keli čistokrvna Kalifornijanka, prvi uslov za budućeg izabranika je da zna engleski.

E, sad ako znate dotični jezik i ako vas interesuje visoka, vrlka mlada plavojka, se pozamašnim kontom i nekoliko kuća i automobila, pogledajte njene uslove, pa ako ih ispunjavate, javite se redakciju za dalji kontakt.

Looking for a Husband

Hardware Requirements

1. Pleasant screen and durable chassis
2. Hard drive – no floppies!
3. User definable sex drive
4. Must come certified „bug free"
5. Top-of-the-line joystick.

Software Requirements

1. Compiler for IOLH (International Obedient Husband Language) basic command subset (e.g. of_course_dear, yes_I_will_watch_the_kids, just_use_the_credit_card)
2. Must be easily programmable
3. Single-user mode ONLY.
4. Very-user-friendly interface

Seka Bajt

UČIMO OD PRIRODE

Genetski algoritmi privlače pažnju programera jednostavnošću osnovnih principa na kojima se zasnivaju. Jednostavan princip daje i jednostavnu realizaciju, uz zavidnu efikasnost. Posebnu draž ovoj metodi daje povezanost sa procesima genetike i selekcije koji se odvijaju u prirodi. Konačno, genetski algoritam se pokazao uspešnim u brojnim primenama.

Đorđe Janković

Genetski algoritam je procedura za nalaženje ekstremuma (minimuma ili maksimuma) funkcija više promenljivih bazirana na analogiji Postupak traženja ekstremuma baziran je na mehanizmima prirodne selekcija i genetike Do ekstremuma se dolazi pretraživanjem po zadatom (višedimenzionom) prostoru tako što se populacija tačaka kreće kroz njega Pri tom uspešnije tačke tj one koje su bliže ekstremu „preživljavaju" a one manje uspešne „izumiru"

PRIRODNA ZASNOVANOST

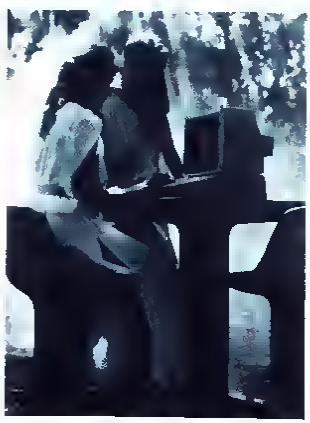
Osnovni cilj genetskih algoritama koji oni uspešno dostižu je robustnost Robustnost je osobina algoritma koja označava mogućnost efikasnog funkcionisanja i u okolnostima koje su različite od onih za koje je on projektovan Prirodni sistemi su znatno robusniji od bilo kog veštačkog sistema Tvorci veštačkih sistema bilo računarskih bilo mašinskih ili poslovnih, mogu samo da sa dive robustnosti, efikasnosti i fleksibilnosti prirodnih sistema Osobine kao što su samopopravljanje samoučenje i samo reprodukcija su sastavni deo prirodnih sistema dok ih malo koji veštački sistem poseduje Pokušaj kopiranja jednog u prirodi prisutnog mehanizma sloga može biti obečavajuće strategija

Tehnika genetskih algoritama zasniva se na osnovnim procesima formiranja novih jedinki i prirodna selekcija Tri operacije su u osnovi procesa selekcije koji se odvija u prirodi

1 „jače" jedinke tj one bolje prilagođene okolini preživljavaju i imaju veći uticaj na formiranje budućih pokoljenja
2 jedinke jedne generacije formiraju novu generaciju i to tako da se osobine novih jedinki dobijaju kombinacijom genetskog sadržaja roditelja
3 ponekad dolazi do mutacije tj do slučajne izmene genetskog sadržaja neke jedinke

OSNOVE ALGORITMA

Zamislimo jednu konferenciju na kojoj vrhunski stručnjaci razmenjuju svoja iskustva o tome kako da reše jedan problem Oni koji su u rešavanju najdalje odmakli najinteresantniji su sagovornici i njihova ideja su najzanimljivija ostalim učesnicima Po povratku sa konferencije u tišini svojih laboratorija svi pokušavaju da kombinuju saopštenu metodu sa zanimljivim idejama ličnih metoda koje su čuli na konferenciji Ideje iz dolada najuspešnijih metoda najviše će se kombinovati Neke od novih kombinacija će biti više a neke manje uspešne Novi rezultati se saopštavaju na novoj konferenciji i proces se odvija dok se ne dođe do rešenja



Slično izloženom primeru funkcioniše i genetski algoritam Do rešenja se dolazi sukcesivno analizom aktuelne populacije tačaka te prevođenjem te populacija u novu populaciju sa istim brojem jedinki Početna populacija se dobija slučajnim generisanjem

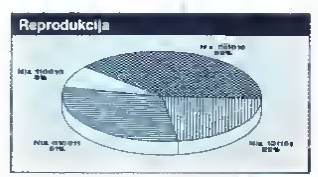
Pojedine tačke (jedinke populacije) predstavljene su nizovima koji su rezultat kodovanja parametara pretraživanja tj varijabli funkcije čiji ekstremum tražimo Parametri se mogu kodovati na različite načine Jedan od najjednostavnijih i najrasprostranjenijih načina kodovanja je binarni U tom slučaju se opseg promene vrednosti svakog parametra podeli na 2^k ($j=2^k$) podopsega Na taj način se svaki parametar može kodovati sa k cifara Te se svi parametri kodiju nizom dužine $k = \Sigma k_i$ Pogledajmo to na jednom primeru Zamislimo da tražimo ekstremum funkcije $u=f(x,y,z)$ Opsezi parametara pretraživanja su za $x - (x_1, x_2)$ za $y - (y_1, y_2)$ i za $z - (z_1, z_2)$ Pretpostavimo da opseg (x_1, x_2) podelimo na 64 podopsega opseg (y_1, y_2) na 128 podopsega i opseg (z_1, z_2) na 32 podopsega Ako konstimo binarno kodovanje opsega onda x podopsege kodujemo sa 6 binarnih cifara (000000 000001 – 111111) y podopsege sa 7 binarnih cifara i z podopsege sa 5 binarnih cifara U skladu sa tim svaka jedinka populacije biće predstavljena nizom binarnih cifara dužine 18 (6+7+5) Jasno je da za svaku tačku (x,y,z) možemo odrediti niz koji je kodiuje a isto tako za svaki niz znamo koju tačku on predstavlja. Pogledajmo koju tačku predstavlja niz 010110101101001110 Ovaj niz delimo na tri dela varijabli x odgovara prvih 6 cifara niza tj niz 010110 (oekadno 21) varijabli y sledećih 7 cifara tj niz 1011010 (90) i varijabli z niz 01110 (14) Tačka koja odgovara ovom nizu je (x,y,z) gde je x=x1+(x2-x1)/64*21 y=y1+(y2-y1)/128*90 i z=z1+(z2-z1)/32*14

Postupak traženja ekstremuma odvija se sukcesivnim ponavljanjem tri osnovne operacije kojima se aktuelna populacija tačaka prevodi u novu

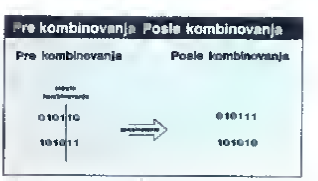
1 Reprodukcija
2 Kombinacije
3 Mutacija

Reprodukcija je postupak kojim se dva i dva niza kombinuju formirajući dva nova niza Sada i nadalje podrazumevaćemo da je cilj pretraživanja pronaći maksimum funkcije f Potpuno analogan postupak je i kada se traži minimum funkcije samo što umesto sa vrednostima funkcije operišemo njihovim recipročnim vrednostima Koji će se nizovi reprodukovati zavisi od vrednosti funkcije u tački koju predstavlja posmatrani niz Tako o vrednostima funkcije možemo razmišljati kao o meri prilagođenosti posmatranog niza okolini Reprodukcija nizova koji imaju veći stepen prilagođenosti je veštačka verzija prirodne selekcije – Darvinove borbe za opstanak među nizovskim bićima U našem veštačkom sistemu vrednost funkcije je sudija koji odlučuje o tome da li će neki niz preživeti ili umreti

Princip reprodukcije se algoritamski vrlo jednostavno realizuje Za svaki niz odredi se vrednost funkcije Vrednosti funkcija svih nizova se saberu i odredi se doprinos svakog niza zajedničkom zbiru U tabeli 1 prikazana je populacija koja se sastoji od čak 10 nizova kao i odgovarajuće vrednosti funkcije Zbir vrednosti funkcije za sve jedinke populacije je 12 38 Doprinos zbiru (DZ) pojedinih nizova se u našem primeru kreće od 8 08% do 39 1 % Izbor niza koji će se reprodukovati možemo sada ilustrovati na sledeći način Zamislimo ruletkolo ima slotove nejednake širine i to tako da su širine slotova proporcionalne vrednostima DZ Primer ruletskog točka za nizove iz tabele 1 pri-



Slika 1 Izbor niza za kombinovanje



Slika 2 Kombinovanje nizova

Tabela 1

Broj	Sekvenca	Funkcija	Doprinos	Kumul.	Opseg
1	011010 (2.6)	4.84	39.10	39.10	1 - 39
2	110010 (5.0)	1.00	8.08	47.18	40 - 47
3	010011 (1.9)	3.79	30.61	77.79	48 - 78
4	101101 (4.5)	2.75	22.21	100.00	79 - 100
Ukupno		12.38	100.00		

Populacija prve generacije

Tabela 2

Broj	Sekvenca	Funkcija	Doprinos	Kumul	Opseg
1	011011 (2.7)	4.91	28.85	28.85	1 - 29
2	010010 (1.8)	3.56	20.92	49.77	30 - 50
3	011101 (2.9)	4.99	29.32	79.09	51 - 79
4	101010 (4.2)	3.56	20.92	100.00	80 - 100
Ukupno		17.02	100.00		

Populacija druga generacija

kazan je na slici 1. Kada završimo rulet, verovatnoća da kuglica upadne u određeni slot proporcionalna je širini slota. Navedeni rulet algoritamski se realizuje na sledeći način.

Formira se kumulativna funkcija DZ koju ćemo označavati sa KF. KF za prvi niz jednaka je vrednosti DZ za taj niz, a za sledeće nizova KF se računa kao zbir KF za prethodni niz i DZ za posmatrani niz. Osobina kumulativne funkcije je da je njena vrednost za poslednji niz jednaka 100. U skladu sa izračunatim vrednostima KF za svaki niz se definiše opseg KF, koji taj niz pokriva. Ganaršamo slučajni broj a u opsegu 1-100. Pronađemo opseg koma pripada a. Niz koma pripada izabrani opseg će se reprodukovati u sledeća pokolenja. Rulet vrtimo tj. generišemo slučajna brojeve onoliko puta koliko ima članova populacije. Očigledno je da kuglica može da upadne u isti slot više od jedanput, tj. jedan niz može da se reprodukuje više no jedanput.

Pošto su izabrani nizovi koji se reprodukuju određuju se parovi nizova koji će se međusobno kombinovati. Ovaj izbor se vrši korišćenjem generatora slučajnih brojeva. Pri tome se ne dozvoljava da se niz kombinuje sa samim sobom. Kada se odrede parovi nizova koji se kombinuju pralazi se na samo kombinovanje. Par po par i to na sledeći način generiše se slučajan broj s u opsegu 1-(k-1) gde je k dužina niza. Na taj način se obadva niza dele na dva dela: podniz 1 koji se sastoji od alamenata 1 do s i podniz 2 koji se sastoji od alamenata (s+1) do k. Nizovi nove generacije se formiraju tako što se prvi niz dobija spajanjem podniza 1 prvog roditelja i podniza 2 drugog roditelja, a drugi niz spajanjem podniza 1 drugog roditelja i podniza 2 prvog roditelja. Tako, na primer, ako kombinujemo nizove 010110 i 101011 uz slučajan broj s=4 kombinovanja se vrši na način prikazan na slici 2.

Operacije reprodukcije i kombinacije su osnovne operacije na kojima je baziran genetski algoritam. Preostala treća, operacija – mutacija igra sekundarnu ulogu. Razlog za uvođenje operacije mutacija je što se može desiti da se primenom dva izložena postupka neki bitan genetski materijal (deo niza) nepovratno izgubi. U genetskom algoritmu mutacija štiti od takvog gubitka. Mutacija je postupak kojim se se određenom (malom) verovatnoćom izabranom elementu niza menja vrednost. U slučaju binarnog kodovanja to znači da će ukoliko izabrani bit ima vrednost 0 njegova nova vrednost biti 1, i obrnuto. Verovatnoća mutacije je standardno oko 0.4 %, što znači da se u proseku sviši jedna mutacija na hiljadu bita. I u prirodnim sistemima do mutacije dolazi sa sličnom ili manjom verovatnoćom.

ALGORITAM U PRAKSI

Način funkcionisanja genetskog algoritma ukratko ćemo razmotriti na jednom primeru. Pretpostavimo da želimo da nađemo maksimum funkcija f(x)= x^2+6x-4. Radi jednostavnosti primera pretpostavimo da je opseg u kome tražimo maksimum x ∈ (0,64) i da taj opseg podelimo na 64 podopsega od po x=0.1. U praktičnoj primeni algoritma broj podopsega je značajno veći. Podopsezi se mogu kodovati sa k=6 (2^6=64) binarnih cifara. Korišćenjem generatora pseudoslučajnih brojeva generišemo na primer 4 (u praktičnoj primeni taj broj je veći) početne sekvence. To generisanje možemo izvesti na taj način što redom generišemo 24 (6*4) slučajna broja u opsegu 1-100. Svaki slučajni broj s određuje po jedan bit niza i to tako da ako je s>50 bit ima vrednost 1, a inače ima vrednost 0. Pretpostavimo da smo dobili sekvence prikazane u drugoj koloni tabele 1.

Za svaku sekvencu izračunata je odgovarajuća vrednost argumenta funkcije x (kolona 2 u zagradi) kao i same funkcije f(x) (kolona 3). Dobijene vrednosti funkcije su sabirane (12.38), te je zatim za svaku sekvencu određen njen udeo u zbiru. Tako je na primer za sekvencu 110010 (dekadno 50) argument x=0 +(6 4-0)/64*50 =5 0 , a vrednost funkcije je f(5 0)=1 00 , te je udeo u ukupnom zbiru jednak 1.00/12.38*100=8.08%.

Sledeći korak je da se odredi koje sekvence će učestvovati u formiranju sledeće generacije sekvenci. U tom cilju generišemo četiri slučajna broja iz opsega 1-100. Pretpostavimo da smo dobili slučajne brojeva 30, 62, 90 i 5. Kao rezultat prvog slučajnog broja 30 izabrana je sekvenca 1 (1≤ 30≤39) kao rezultat drugog broja sekvenca 3. Ireći – sekvenca 4, a četvrtog opet sekvenca 1. Znači, sekvence 1 će učestvovati u reprodukciji dva puta, a sekvenca 3 i 4 po jedanput. Vidimo da se sekvenca koja predstavlja segment koji je najbliži maksimumu reprodukuje dva puta.

Sledeći korak je da se vidi koja će se sekvence kombinovati. Upotrebljene sekvenci koje će se kombinovati vrši se isto tako korišćenjem generatora slučajnih brojeva. U našem slučaju jasno je da se sekvenca 1 kombinuje se sekvencama 3 i 4, zato što se sekvenca ne može kombinovati sama sa sobom. Kombinujemo prvo sekvence 1 i 3. Treba da odredimo mesto na kom će se sekvence kombinovati. Generišemo slučajan broj iz opsega 1-5 (5=k-1, gde je k=6 dužina sekvence). Pretpostavimo da je dobijen slučajan broj 4. Sekvence 1 i 3 se kombinuju tako što se biti 5 i 6 sekvenci 1 i 3 međusobno izmanjuju.

0110 | 10	0110 11
0100 | 11	0100 10

Na sličan način se kombinuju i sekvence 1 i 4, uz slučajan broj koji određuje mesto kombinacija – 2.

01 | 1010	01 1101
10 | 1101	10 1010

Po obavljenoj kombinaciji generišemo se slučajni broj da bi se videlo da li se viši mutacija. Opseg slučajnih brojeva je 1-1000 a ukoliko se dobije broj jedan viši se mutacija. Kako je verovatnoća mutacije vrlo mala pretpostavimo da se u ovom koraku neće vršiti mutacija.

U tabeli 2. date su vrednosti funkcije za nove sekvence. Može se primetiti da je zbir vrednosti funkcije u drugoj generaciji populacije povećan sa 12.38 na 17.02, a pronađeni maksimum sa 4.84 na 4.99.

Pretraživanja se nastavlja izloženim postupkom dok se ne dođe do zadovoljavajućeg rešenja. Kao i kod drugih algoritama pretraživanja, kriterijum završetka može biti različit: kada je u dve uzastopne iteracije poboljšanje (bilo celokupne populacije bilo tačka najbliže maksimumu) manje od graničnog, kada se sve tačke populacije nalaze una međusobnom rastojanju manjam od graničnog, itd.

NJEGOVE PREDNOSTI

Osnovne karakteristike genetskih algoritama koje ih razlikuju od drugih metoda pretraživanja i iskazuju se kao njihova prednost su sledeće:

1. Genetski algoritmi operišu sa jednim nizom koji se dobija kodovanjem svih varijabli, a ne rada sa pojedinim varijablama.
2. Rešenja se dobija ne pomeranjem jedne tačke po domenu pretraživanja već pomeranjem skupa (populacije) tačaka. Ovo je jedan od osnovnih razloga robustnosti genetskih algoritama. Ukoliko se do rešenja dolazi pomeranjem jedne tačke u slučaju funkcija sa više lokalnih maksimuma, lako se može desiti da se tačka veže za neki lokalni maksimum. Nasuprot tome genetski algoritam radi sa skupom tačaka koje se istovremeno penju ka različitim maksimumima. Na taj način se smanjuje verovatnoća pronalaženja lažnog maksimuma.
3. Pri evaluaciji uspešnosti jednog niza tj. jedne tačke pretraživanja, koristi se samo podatak o vrednosti funkcije u posmatranoj tački, a ne izvod funkcije u toj tački ili neka druga informacija o prirodi i ponašanju funkcije. Ova osobina genetskih algoritama kvalifikuje ih za primenu u širokom spektru problema.
4. Pri generisanju nove populacije tačaka koriste se probabilističke (zasnovana na slučaju), a ne determinističke, pravila. Međutim kao što smo videli iz opisa algoritma, slučajnost pri prelazu iz jedne populacije u drugu nije slepa, već kontrolisana i upravljana slučajnost.

Na ovom mestu izložili smo samo osnovne genetskih algoritama, u žalji da zainteresujemo čitaoce za ovu nama vrlo interesantnu metodu. Mnogi aspekti nažalvno nisu mogli biti razmatrani teorijska zasnovanost metode, detalji realizacije, raznovrsne primene, napredne tehnike. Za čitaoce koji su zainteresovani za metode genetskih algoritama, na žalost, na našem jeziku ne postoji knjiga posvećena ovoj problematici, ali ih možemo uputiti na vrlo dobru knjigu Davida Goldberga *Genetic Algorithms in Search, Optimization and Machine Learning* u izdanju Addison-Wesley.

ŽIVOT JE LEP!

Istorija, pa i istorija računara, se stalno ponavlja. Da li je konačno došlo vreme da se oživi san Džona fon Nojmana o ćelijskom automatu?

Miroslav Olenjin

Igra-simulacija LIFE (život) vuče korene još iz radova čuvenog Džona fon Nojmana na takozvanom „ćelijskom automatu" On je od 1948 pa sve do svoja smrti 1957. godine proučavao mogućnost izrade mašina koja bi se odlikovale sposobnošću regenerisanja i samo replikiranja – osobinama svojstvenim jedino živim bićima. Došao je do fundamentalnog zaključka da bi se jedna takva mašina morala sastojati od velikog broja jednostavnih i uzajamno zamenljivih elemenata sličnim organskim ćelijama. Svaki bi pri strogo određenim uslovima trebao da proizvede sebi identičan element što bi na višem nivou rezultiralo drugom grupom elemenata u istovetnoj konfiguraciji.

OBNOVLJENI SAN

Fon Nojman se nije, znajući da mu ondašnje vreme to i ne omogućava, bavio konkretnom izradom takve masine. Njega je više zanimao kupljala misao da dođe do univerzalnog rešenja... do genalskog koda s kojim bi na papiru dokazao mogucnost njenog funkcionisanja. Ćelijski automat koji je on zamislio se sastojao od jedne beskrajne ravni podeljene na kvadratiće svaki kvadrat bi se nalazio u jednom od 29 moguca stanja dok su pravila prelaza iz jednog stanja u drugo bila veoma slozena. Uz to svaki početni organizam bi se gradio od više podsklopova koji imaju sposobnost samoreplikiranja. Iako da bi njihovo umnožavanje dovodilo i do umnožavanja celog organizma. Na ovaj način je navodno moguće kreirati organizam bilo kakvog oblika a da on zadrži sposobnost samoreplikiranja! Da li je fon Nojman u tome uspeo ni do danas se ne zna...

U narednim decenijama ideja o samoumnožavajućoj mašini gubi značaj ali se za koncepcije ćelijskih automate sve više zanimaju konstruktori računara. Postajući svesni ograničenja klasičnih kompjutara sa jednim centralnim procesorom okreću se paralelnim arhitekturama koje podržavaju simultan rad mnoštva procesora, međusobno gusto povezanih popul mreže neurona. Za njihovo projektovanje su obilato korišćena dostignuća Fon Nojmana, Lengtona Berksa, Koda i drugih lapreticara ćelijskih automata – s tom razlikom da je prostiranje i obrada informacija kroz neuronsku mrezu analogno umnožavanju ćelija.

Sam LIFE kojim ćemo sa mi baviti pronašao je Džon Horton Konvej sa Univerziteta u Kembridžu 1970. godina. Simulacija se odvija na kvadratnoj tabli: ako je u pitanju dvodimenzionalni lajf odnosno u prostoru oblika pravougle prizme ako je reč o trodimenzionalnom. Svaka pozicija (polje) u okviru log radnog prostora može imati dva stanja, sadržavati živu ili mrtvu ćeliju. U stalnim ciklusima se stvaraju nova generacija ćelija što simboliše njihovo biološko razmnožavanje. Sudbina jedinka zavisi od njene okoline: da bi neka živa ćelija preživela ciklus mora biti okružena sa najmanje **n** a sa najviše **m** drugih inače umire. Prazna pozicija, pak, može oživeti pod uslovom da je okružuju najmanje **p** ali najviše **q** živih pozicija. Brojevi **n**, **m**, **p** i **q** su od presudnog značaja za odvijanje simulacije jer određuju pravila života – isti početni organizmi skupovi ćelija koji se zadaju pre startovanja simulacije neće se razvijati identično pod različitim pravilima. Kod novije verzije lajfa pozicije oko neke žive ili mrtve ćelije se ispituju i dijagonalno. Iako da se njena okolina kod dve dimenzija sastoji od 8 (kao što ide kralj u šahu) a kod tri dimenzija od 26 polja (Rubikova kocka). Otuda je trodimenzionalna simulacija znatno zahtevnija prema računaru, tako da za nju i nema smisla koristiti ništa što ima manje od 16 bita.



ALGORITAM

Lajf se, bez obzira na varijantu, svodi na sledeće. Definišu se u memoriji računara dva matrice, jedna za čuvanje sadašnje a druga za izračunavanje naredne generaciju. Na osnovu rasporeda ćelija u prvoj matrici izračunava se njihov broj i raspored u sledećoj generaciji (pri manjujući unapred zadata pravila) I paralelno upisuje u drugu matricu. Polom sa druga prekopira u prvu, rezultat se iscrtava na ekran pa se onda ceo poslupak ponavlja.

Kao uostalom i uopšte u programiranju, i ovde postoje jednostavniji/sporiji i komplikovaniji/brži algoritmi da bi se ovo postiglo. Najprostije bi bilo ići od jedne pozicije do druge brojati žive ćelije u njihovom neposrednom okruženju, pa zatim u zavisnosti od njihovog statusa odlučivati o njihovoj budućnosti: ako je to živa ćelija, onda prema pravilu o opstanku a ako nije onda prema pravilu o rađanju. U svakoj iteraciji bi se znaci ispitivale sve pozicije. Dalako je međutim efikasnije da se matrica pretražuje u potrazi samo za živim ćelijama, pa kada se nalazi na naku da se pored loga što će se razmotriti njeno preživljavanje ispita i mogućnost oživljavanja nekog od okolnih mrtvih polja. Jer, treba to uočiti, kandidate za oživljavanje valja tražiti jedino u blizini postojećih živih ćelija, zbog poslojanje uslova o njihovom minimalnom broju u okolini prazne pozicije. Kako se sa ovim algoritmom ispituje mnogo manje njih, to je on nekako brži od prethodnog. Razlika u brzini je posebno primetna u prvih nekoliko iteracija kada još nema mnogo živih ćelija.

Ipak, i ovaj postupak nije savršen: većina praznih polja se ispituje više puta. Tačnije, onoliko puta koliko ih živih ćelija okruzuje – tako da bi se teoretski jedna pozicija mogla ispitivati i po 26 puta! Naravno rezultat će biti stalno isti. Zato uvodimo jednu dodatnu matricu istog formata kao i dve glavne, u koju ćemo beležiti da smo određenu praznu poziciju već jedanput proveravali. Ta treća matrica će nam lakoda posluziti i da spričimo „izlazak" organizma izvan radnog prostora. To znači da se na sme dozvoliti da u ivičnim pozicijama bude živih elemenata, jer bi onda pri ispitivanju njihove okoline došlo do ispadanja iz deklarisanog područja. Njih neće biti ako ne dopustamo prazni granični položaja da oživljavaju – a ovo cemo postići tako sto cemo ih u trećoj matrici označiti kao već ispitivane. Time doduše praktično ostajamo bez celog periferno područja, ali zato minimalno gubimo na brzini!

SA REČI NA DELO

Prema ovom algoritmu napisan je, na Turbo Pascalu, program LIFE3D za simulaciju tro-

Listing 1 Life 3D

```
program LIFE3D;
{
      Simulacija trodimenzionalnog lajfa
      Miroslav Glanjin (c)  avgust 1991
}
uses  Graph, Crt;

type  Matrice = array [0..25, 0..25, 0..26] of Boolean;

var   x, y, z                 : byte;
      p, q, r                 : byte;
      v, t, s                 : byte;
      MINSURV, MAXSURV        : byte;
      MINBORN, MAXBORN        : byte;
      Okolina1, Okolina2      : byte;
      Gd, Gm, Gen             : integer;
      prompt1, prompt2        : string;
      a, b, c                 : Matrice;


procedure UnosKoord;
begin
 for z:=0 to 26 do begin
  for y:=0 to 26 do begin
   for x:=0 to 26 do begin
    a[x,y,z]:=false;
    b[x,y,z]:=false;
    c[x,y,z]:=true;
   end;
  end;
 end;
 write ('Najmanje za opstanak : '); readln (MINSURV);
 write ('Najvise  za opstanak : '); readln (MAXSURV);
 write ('Najmanje za radjanje : '); readln (MINBORN);
 write ('Najvise  za radjanje : '); readln (MAXBORN);
 writeln;
 repeat
  write ('Unesite vrednosti koordinata: '); readln (x,y,z);
  if (x>0) and (y>0) and (z>0) then b[x,y,z]:=true;
 until (x=0) or (y=0) or (z=0);
end;


procedure Iscrtajsliku;
var   SetX, SetY, xl, yl, Celije : integer;
begin
 SetViewPort (0,0,480,470,true);
 ClearViewPort;
 SetX:=140;
 SetY:=320;
 Celije:=0;
 for z:=1 to 25 do begin
  Dec(SetX, 5);
  Inc(SetY, 5);
  for y:=1 to 25 do begin
   for x:=1 to 25 do begin
    a[x,y,z]:=b[x,y,z];
    c[x,y,z]:=false;
    if a[x,y,z] = true then begin
     Inc(Celije);
     xl:=SetX+(x*11);
     yl:=SetY-(y*11);
     for p:=0 to 9 do begin
      setcolor(1); line(xl+10, yl-p, xl+13, yl-p-3);
      setcolor(3); line(xl+p, yl-9, xl+p+4, yl-13);
      setcolor(0); line(xl+p, yl, xl+p, yl-9);
```

```
     end;
    end;
   end;
  end;
 end;
 SetViewPort (450,350,630,450,true);
 ClearViewPort;
 Rectangle (0,0,180,100);
 Str (Gen, prompt1);
 Str (Celije, prompt2);
 OutTextXY (55,  15, 'LIFE - 3D');
 OutTextXY (20,  50, 'Generacije : ');
 OutTextXY (20,  75, 'Broj celija: ');
 OutTextXY (130, 50, prompt1);
 OutTextXY (130, 75, prompt2);
end;


begin {progrsa}
 repeat
  ClrScr;
  Window (20, 5, 60, 20);
  UnosKoord;
  Gd:=Detect; InitGraph (Gd, Gm, '');
  Gen:=0;
  repeat
   Inc(Gen);
   Iscrtajsliku;
   for z:=1 to 25 do begin
    for y:=1 to 25 do begin
     for x:=1 to 25 do begin
      if a[x,y,z] then begin
       Okolina1:=0;
       for r:=z-1 to z+1 do begin
        for q:=y-1 to y+1 do begin
         for p:=x-1 to x+1 do begin
          if a[p,q,r] then
           Inc(Okolina1);
         end;
        end;
       end;
       if a[p,q,r] = false then begin
        Okolina2:=0;
        for s:=r-1 to r+1 do begin
         for t:=q-1 to q+1 do begin
          for v:=p-1 to p+1 do begin
           if a[v,t,s] then Inc(Okolina2);
          end;
         end;
        end;
        if (Okolina2 >= MINBORN) and (Okolina2 <= MAXBORN) then
         b[q,q,r]:=true else b[q,q,r]:=false;
       end
       else
        c[p,q,r]:=true;
      end;
     end;
    end;
    Dec(Okolina1);
    if (Okolina1 >= MINSURV) and (Okolina1 <= MAXSURV) then
     b[x,y,z]:=true else b[x,y,z]:=false;
    end;
   end;
  end;
  until KeyPressed;
  CloseGraph
 until ReadKey=#27
end   {progrsa}
```

Program na Turbo Pesceiu za simuleciju trodimenzionainog života

dimenzionalnog života. Radi u grafičkom modu 640 x 480 VGA kartice, no čitaoci koji imaju EGA ili Hercules neće imati velikih ješkoća da ga prepravi. Veličine radnog prostora ograničene je na 25 x 25 x 25 odnosno na zapreminu od 15625 pozicija. To je kompromis između želja da se naš organizam što slobodnije razvija i mogućnosti računara da podrži simulaciju u realnom vremenu. Ne treba zaboraviti na me. da će se vreme odziva povećati osam puta ako udvostručimo dimenzije radnog prostora. Deklarišu se tri matrice (**a**, **b** i **c**) čiji su elementi tipa Boolean, kako bi se postiglo što brže manipulisanje sa njima. Pošto su matrice dimenzija 27 x 27 x 27, zauzeće ukupno 59049 bajtova, što zajedno sa ostalim varijablama na prelazi 64 K, pa nam svi podaci stoje u jednom segmentu.

Nakon što ste ga pažljivo uneli u računar i pokrenuli program će od vas tražiti da unesete pravila i koordinate ćelija koje će činiti početni organizam. Koordinate početak se nalazi u udeljenjem donjem desnom uglu. Najbolje da unosite po tri broja odjednom, odvajajući ih blankom. Obratite pažnju da oni budu u opsegu 1 25. Ie da z koordinata na ekranu definiše „dubinu" a ne „visinu". Možda ovo na prvi pogled na izgleda logično, no pokazalo se boljim da se poštuje navika kretanja po ekranu pomoću miša ili kursorskih tastera. Unos se završava kada nekoj od koordinata dodalite nulu. Tada program prelazi u grafički režim i prikazuje početni oblik. Ćelije su predstavljene u vidu kockica u kosouglo projekciji i iscrtavaju se tako da one bliže posmatraču prekrivaju one udaljenije. Računar će stvarati nove (nove generacije) i u donjem desnom uglu ekrana ispisivati njihov redni broj i broj živih ćelija sve do pritiska na neki taster. U slučaju da je to bio ESC, izlazi se iz programa inače se kreće iz početka.

Ovaj kratak listing bi predstavljao samo srž većeg programa koji možete dopisati oko njega. Poželjno je recimo, dodati opis pet-šest različitih početnih organizama kako se ne bi stalno ponavljao dosadni postupak unošenja koordinata. Bilo bi zanimljivo i gledati na ekranu paralelno razvoj generacija i videti kako je bilo potrebno za njihovo generisanje. Ili dodati nom rutinom u asembleru za iscrtavanje kockice ubrzati iscrtavanje koja ovako traje duže nego samo računanje? A opasnijim programerima (i svima koji se lako osećaju) preporučujem razvoj procedura koje bi same menjale pravila i početne oblike i automatski upravljala celokupnim programom u potrazi za atraktivnim organizmima (!) O kojima će sada biti reči.

PRAVILA IGRE

Ne startu se promenljivim MINSURV, MAXSURV, MINBORN i MAXBORN dodeljuju vrednosti za najmanji i najveći broj živih ćelija potrebnih da okružuju ćeliju da bi ona opstala odnosno da bi prazna pozicija oživela. Promenom ovih konstanti menjaju se, dakle, i sama pravila igre od njih zavisi brzina i način na koji će se početni organizam razvijati, da li će biti plodan i za kratko vreme preplavljivati ceo ekran ili će brzo „lunjav" i nestajati nakon par generacija. Upornim eksperimentisanjem se mogu naći i zaista izuzetno zanimljiva pravila pri kojima se javljaju mnogi stabilni organizmi čija je osobina da zadržavaju svoj izgled – pod uslovom jasno, da u njihovoj neposrednoj blizini nema drugih živih ćelija – i razne kreature koje u toku dva ili više ciklusa ponavljaju svoje oblike pri tom se čak i krećući u određenom smeru. Lajf je pun iznenađenja.

Ako pn pravilu 4-4-4-4, na primer, definišete figuru u obliku pločice sa koordinatama

```
13 13 13
13 13 12
12 13 13
12 13 12
```

videćete kako će ona narasti do jednog posebnog organizma koji ima sposobnost samoreplicianja. Dok isla pločica pn pravilu 2-4-4-6 daje lek niz oktaedara. Podite sada iz ugla sa početnim oblikom:

```
1 1 1
1 1 2 ,
1 2 1
2 1 1
```

I pravilom 2-3-1-1 i videćete prekrasnu igru života koja se ponavlja u periodima od desetak generacija. Sasvim druga priča je kod pravila 2-6-4-6 ili 4-5-4-5. Raznih kombinacija nema bezbroj, ali ih ima taman toliko da svako ko je dovoljno uporan napravi neko sopstveno otkriće!

POTPIŠI SE I .. POČNI

Baveći se različitim tipovima hardverske zaštite programa, Vlada Kostić je u prošlom broju pokazao sve prednosti i mane ovakvih aistema. U ovom tekatu će biti reči o drugačijem pristupu istom problemu.

Zoran Živetić

Zaštita programa od neovlašćenog kopiranja je večna teme. Večna iz prostog razloga što još nije pronađen pouzdan sistem kojim bi se ona obavila. Tenjski gledano, ne postoji (bar za sada poznat) način kojim bi se program mogao potpuno zaštiti. Zaštita se može učiniti komplikovanom i teškom za razbijanje ali će uvek postojati način da se ona skine. Pored klasičnih hakerskih maloda koje traže dosta vremena i lskustva razbijačima uvek stoji na raspolaganju i ozbiljniji (mada skup) alat kojim se i najkomplikovanije zaštita može veoma brzo ukloniti. Nevolja je, naravno, što je sasvim dovoljno da se zaštite razbije samo na jednom primerku – nakon toga trud i novac uložen u hardverske dodatke svih ostalih kopija postaje besmislen.

TVRDI HARDVER...

Hardverski dodaci ostaju ipak nezamenjivi kod skupih uskom profesionalnom tržištu namenjenih programa. Na ovom segmentu problem kopiranja je nešto manje izražen (mada je i šteta od jedne dodatne kopije veća) već samim tim što program podrazumeva i stvari koje se teško ili nikako ne mogu kopirati, kao što su obimna dokumentacija i podrške od strane proizvođača.

Drugi segment tržišta – jeftiniji programi opšta namena donosa proizvođačima daleko veće glavobolje. Kako uz program koji košta recimo 100 USD ili manje i za koji se očekuja tiraž od više desetina ili slotina hiljada primeraka priključili bilo kakvu hardversku zaštitu? Cena programa bi morala biti nerealno visoka, programa ovog tipa svaki korisnik ime više na svom računaru pa bi se dodaci nagomilali do nepodnošljivih razmera, što bi dalje smanjilo potražnju. A opet, ovakvi programi se najčešće neovlašćeno kopiraju i to bas na onaj najmasovniji način – prijateljskom razmenom.

Proizvođači u poslednje vreme konste lehniku koja očigledno ima osloneca i u poznavanju psihologije. Zamislimo sledecu situeciju: kupili sla program po ceni od 50 USD koji nema ni kakvu zaštitu od kopiranja. Program neme ni kakav serijski broj, pa ukoliko ga dale prijatelju neće postojati način da bilo ko sazna odakle ga je dobio. Doduše, uvak se kod vas može javiti sumnja da je vaša kopija ipak nekako obaležena, ali s obzirom de namale ocigledno dokaza za to, uvek će pobediti osećaj zadovoljstva zbog zahvalnosti onoga kome činite uslugu. Osecaj vlasništva u ovom slučaju se očigledno svodi samo na stvari koje čine program (moje uputstvo, moja diskela), ali ne i na sam program koji se izvršava na računaru.

Proizvođači su odavno konsili lehniku obeležavanja kopije serijskim brojem, ali je to uvek činjano na neki skriven način. Niko se do ne davno nija seto da pored broja zahteva od vas da kopiju obalezita i svojim imenom. A ta da je situacija sasvim drugačija. Ako se pri svakom starovanju programa na ekranu pojavi „Ova je lična kopija Pere Perića" broj ljudi koljima cete bili spremni da date program će svakako biti daleko sužen! Osećaj vlasništva se lada širi na sam program. S obzirom da je svasl o naposlenosli deljenja programa sve više prisutna kod korisnika, malo ko će prsiali da sa njegovo ime na lakav način „vuče po računarima". Ako smo na ovaj jednostavan nacin elimisali deo „prijataljskog piratstva", da pogledamo šta je sa onim manje dobronamernim. Poslavljaju se dva pitanja. Kako izazov ovakva zaštita nameće i koliko se lako može skinuti?

...I TVRDE OD HARDVERA

Odgovor na prvo pitanje je veoma inlaresantan. Daleko je veci izazov pronaći način da upišete svoje ime od toga da se prosto izbaci provera da li je ime upisano ili ne! Pored toga, upisivanjem imena „vlasnik" odbija cak i mogućnost da se javno deklariše kao „pošlen" korisnik koji je program kupio. Pirat koji na ovaj način žali da napravi više kopije treba potpuno da shvatim mehanizam po kome se ime upisuje u program i da potom ganansje program kojim će ovo moći da se obavlja „po zahtevu". Ako i jeste izazov proneći kako izazov preslaje u lranutku kada se shvati mehanizam pa ukoliko je program za kodiranje nasio komplikovanu za realizaciju, ponako će odustati od tog posla zaodvoljavajući se jednom ili sa nekoliko kopija koje je uspeo de napravi.

Koliko je teško skinuli ovakvu vrstu zaštile? Odgovor je – teže nago hardverski. Razlog je jednostavan – ne postoji tačka na koju zaštita mora da stigne, jer nema dodira sa hardverom. Mogucnosti programera da „zapelja" stvar su daleko šire. Sučan se samo jednom kopijom obelezenog programa razbijač zaštite praktično nema odakle da počne.

Svakako i ovakvu zaštitu je moguće skinuti – dovoljno je da se uporede dva kopije i to će ojkriti mesla na kojima je udaren pečat. Izmenom podalaka u ovoj oblasli dolazimo u isiu situeciju identičnu oslalim tipovima zastite – treba ispratiti lok programa do mesle na kome uočava neovlašćenu izmenu i spreći ga da izveda za taj slučaj predviden izlazak iz programa. Ipak, kao što rekosmo ovim izazov ne nestaje „Prava stvar" je pronaći način de se ime zameni a ne samo učini nečitkim. To već može biti dalako dalako leži zadalak jer zahteva da se potpuno shvati kako deo programa radi. Ukoliko se registracioni podaci kodiraju potrebno je pronaći način kako su kodirani da bi se moglo našlo smisleno upisali i pored toga pronaci ili nacin na koji je izvedena zaštita ili način kako da se spreci ova provera. Programer koji pravi zaštitu može ovaj posao učiniti toliko teškim da se praktično može smatrati nemogucim.

Ocigledno je da ova zastila predstavlja srećnu kombinaciju za određen lip programa. Ona ima i niz za proizvođača praktičnih prednosli. Najčešce primedbe korsnika na razne vrsla zaštila se odnose na nemogucnost pravljenja zaštitnih kopija programa ili na opterecenosl hardvara raznim dodacima. Obe primedbe ovde ne stoje. Realizacija zaštite se najcešće izvodi tako šlo na samom programu pri prodaji nema nikakva oznake. Tek pri prvom starlovanju korsnik sam treba da upiše svoje ime i registracioni broj koji se nalazi na posebnom registracionom kartonu. Zahvaljujuci ovoj lehnici „samoregistracije" od strane korsnika program se može distribuirali na najrazličilje načine. Čak sa može stimulisati i davanje programa poznanicima, jer lak nakon plaćanja i oni dobijaju svoj registracioni karton i priliku da obeleže svoju kopiju. Pored toga ukoliko je firma prisiljene da izbaci novu naku verziju zbog ozbiljnih nedostataka koji su uočenu u lekucoj verziji (sto se onda radi bespiatno), nacin distribucije im je znatno proširen. Mnoge firme konste modemske komunikacije i otvaraju sopstvene servise na kojima svako može potpuno slobodno preuzali ispravnu verziju programa. Naravno, samo oni koji vec imaju svoj serijski broj moći će i da je registruju.

NEOGRANIČENE MOGUCNOSTI

Kao što smo rekli realizacija ovakve vrste zaštite pruža programaru neograničene mogucnosli. Sam algonlam je u osnovi jednostavan. Negde u programu (kao primer konsliće-mo ime PRG EXE) treba ostaviti mesto za upisivanje serijskog broja kopije i imena vlasnika. Ovo podruje treba da ima slatlički definisan sadržaj, po kome se moza prepoznati da kopija još nije registrovane. Posebnim programom za registraciju (neka se zove REG EXE) u taj prostor će biti upisani podaci o konsniku a

Listing 1 Findmarker

```
findmarker(char *fname, char *markertext)
{
    int pos,c;
    long fpos;
    FILE *f;
    pos = 0;
    fpos = 0;
    f = fopen(fname,"rb");
    for ( ; (c=fgetc(f)) != EOF ; ++fpos )
    {
        if ( !markertext[pos] )
            break;
        else if ( c != markertext[pos] )
            pos = 0;
        else
            ++pos;
    }
    fclose(f);
    return( c!=EOF ?  fpos -= strlen(markertext) : -1L );
}
```

Jednostavne funkcija za pronalazenje markera u programu



sadržaj mora biti na neki način zaštićeten tako da se može prepoznati neovlašćena izmena. Prvo je pitanje – kako odvojiti prostor i potom ga pronaći programom za registraciju?

Očigledno rešenje je deklarisati jednu strukturu

```
struct {
    long regnum,
    char regname[32];
    } regdata = { 0L, "Not registered" };
```

Nakon kompajliranja možemo nekim od alata za direktan pregled sadržaja datoteke pronaći ovu sekvencu u EXE datoteci, zapisati ofset na kome se nalazi i potom u program kojim se up suje registarski broj praviti ovu vrednost. Ovakvo rešenje ima puno mana – nakon svake izmene PRG EXE moraćemo da proveravamo da li se adresa strukture promenila da bi adekvatno izmenili i REG EXE. Zato je daleko bolje konstiti metod markera – u program ćemo upisati neki jedinstven string i potom ga programom za registraciju pronaći. Kao marker odlično može da posluži copyright poruka pa možemo našu strukturu proširiti na sledeći način

```
struct {
    char cprght[22],
    long regnum,
    long regname[32]
    } regdata = { "(C) Pera Peric, 1991.", 0L,
"Not registered" };
```

Sada dolazi na red ključno pitanje – kako ovo područje kodirati i zaštiti od izmena? Pre nego što posegnete za literaturom i 'laškim' alatima kao što su računanje CRC a i neprobojni algoritmi za kriptografisanje – razmislite malo i o jednostavnijim i manje očiglednim načinima. Ima ih neograničeno – dali neki konkretan je besmisleno jer time postaje svima poznat pa ga i ne treba primeniti. Ipak, evo jedna od ideja – podatke preskakamo u još jednu kopiju identične strukture ali na neki ispremešan ili drugačije kodiran način. Posle je osnovni problem potreba da se lačno zna gde se ta kopija nalazi jer treba i REG EXE da je pronađe. Možemo uvesti neki drugi manje očigledan marker ili proslo na postojeću strukturu dodati još jedno polja **regnum** i **regname**. Podaci se u ova polja mogu prepisati nekim drugim redosledom manjajući vrednosti. Recimo prvi bajt serijskog broja može biti upisan kao 16 bajt u polju **regname**, oduzimajući prethodno od njega broj 17. Ovo se može obaviti u jednoj proceduri ali ako stvarno želite da „zamulite" možete prebacivanje i proveru protegnuti na više procedura, koje će premeštati i proveravati ispravnost samo pojedinih delova. Naravno važno je da ne napravite klasičnu grešku koja se inače dešava – da program slučajno „zapeljate", a potom na jednom jedinom mestu proveravate da li je došlo do povrede sadržaja registracionih podataka.

Kodiranje podataka takođe može biti izvedeno na neki jednostavan ali na lako očigledan način – nema ni svrhe da u programu od recimo 40-60 K koda 10 ili 20 olpada na zaštitu. Možete ovaj korak i preskočiti – neka razbijač pronađe ime i proba da ga izmeni – ukoliko je prava integritela podataka dovoljno bezbedna od zamena imena će imati malo konsti.

Konačno ostaje još jedno teorijsko pitanje koja treba rešiti – izbor serijskog broja. Naime, broj ne može biti bilo kakav jer bi onda sa parom neregistrovani PRG EXE i REG EXE svako mogao da upiše bilo koje podatke i lako dođe da „legalne" kopije. I ovde važe slične napomene kao i ranije – broj treba da bude formiran tako da samo odreden skup brojeva „ima smisla". Najčešće se to radi podelom broja na dva dela – jedan koji je zaista sam broj kopije i drugi deo koji može biti opet neka vrsta kontrolnog zbira ili kodiran prvi deo i slično. Brojevi zalo nužno moraju imati malo veći broj cifara/slova – leško da se sa manje od 7 ili 8 cifara može napaviti afikasna zaštita. Mozete slobodno i preterati kao što to rade neke firme – serijski broj od palnaestak cifara s obzirom da ga konsnik upisuje samo jednom neće izazvati nikakav problem. Vezno je napomenuti da izbor serijskog broja čini jednu kariku u celom lancu zaštite – ako je ovaj deo posla napravljen loše, čitava zaštita puca.

SPISAK POSLOVA

Evo na kraju i sumarnog pregleda onoga što treba napraviti:

1) Izabrati algoritam za zaštitu dela memorije od izmena podataka koje sadrži

2) Izabrati algoritam za kodiranje sadržaja dela memorije (opciono)

3) Izabrati algoritam za formiranje serijskog broja

Potom u kodiranju PRG EXE

1) U program PRG EXE postaviti marker na područje u koje će biti upisani registracioni podaci

2) U programu PRG EXE kodirati proveru integriteta područja sa registracionim podacima

Zatim kreirati REG EXE koji treba da sadrži sledeće delove

1) Provera ispravnosti unetog serijskog broja

2) Pronalaženje markera u PRG EXE

3) Upisivanja serijskog broja, kodiranje (opciono) i postavljanje ostalih sadržaja koji sluze za proveru integriteta registracionih podataka

Program REG EXE može biti ugrađen i u sam PRG EXE. Sa stanovista sigurnosti, to može biti i bolje jer je nekad lakše analizirati poseban kraci program za koji znate lacnu namenu i sve module koje verovatno sadrzi. S druge strana neracionalno je opterećiti osnovni program kodom koji ce biti izvršen samo jednom.

Iz „uraditi" liste se vidi da nema ni jedne tačke koja bi mogla biti komplikovana za realizaciju. Skrećem vam ipak paznju na lačku 2 pri realizaciji REG EXE. S obzirom da brzina pronalaženja markera nije mnogo bitna, možete se poslužiti jednostavnom funkcijom sa listinga 3.

Funkcija će vratiti -1 ako marker ne postoji ili poziciju prvog znaka markera u EXE dalolaci.

PRAVI PROGRAMERI

KOJI JEZIK KORISTITE

Spektakalaran razvoj računarstva a posebno programskih jezika, doveo je do čudesne situacije: postoje bezbrojni programski jezici koji imaju još bezbrojnija svojstva i sintaksu ukradanu jednu od drugih. Naravno, kvalitet programara se često meri i brojem jezika koji zna, ili bar misli da zna. Tako je i nastala ova knzna situacija u kojoj je programaru sve teže da se razabere tokom pisanja programa koji jezik zapravo konsti.

Da bismo vam bar malo olakšali situaciju, saslavili smo kratko uputstvo koje bez ikakve nadoknade i bez straha od pravničkog progonjanja, može da pomogne programerima i u najtezim situacijama. Jednostavno, kad ne znate u kom jezikku upravo programirate, pogledajte u svoju nogu i značete šta konslite.



C
Efikasno i koncizno si se upucao u nogu.

Asembler
Srušio si bez većih problema operativni sistem i prepisao sistemske programe. Sistem administrator dolazi i upucava te u nogu, zatim upucava sebe u nogu i konačno avakog u vidokrugu upucava u nogu.

C++
Malom greškom kreirao si tuce svojih instanci i sve ih upuceo u nogu. Pružanje prva pomoći je nemoguće zato što sva kopija pokazuju jedna na drugu i govora „To sam ja, baš tamo"

Ada
Ako zaista konstite ovaj jezik američko Pentagon vas je odavno kidnapovao, stojite pred streljačkim strojem (u čijoj pozadini stoji veliki kolor poster nekog od upropašćenih projekata) i izdalo je naredjenje „Pucajte u njegove noge"

Moduta-2
Pošto konačno shvatite da, u stvari zaista nema ni najmanje šanse da bilo šta ozbiljno isprogramirate u dotičnom jeziku, vi se neozbiljno ali uspšno upucavale u nogu.

Smalltalk
Pošlo provedeš čtavo radno vreme igrajući se grafikom i prozorskim sistemima, šel dolazi i upucava te u nogu, oduzima li glanc novu radnu stanicu i nabavlja ti karakter terminal sa isključivo COBOL kompajlerom.

APL
Čuješ pucanj, rupa je u tvojoj nozi, ali se na možeš sešti dovoljno linearne algebre da bi zaista shvatio šta se desilo.

Seka Bajt

PUCANJ KROZ PROZOR

Ovim prilogom o programiranju u C-u za *Windows* počinjemo da koristimo upravo ono zbog čega su se mnogi opredeljivali za *Windows* egzotiku i mnogo pre nego što su „prozori" postali naša svakodnevnica.

Dejan Dinić

Kao i obično, nastojaćemo da konkretnim programskim primerom ilustrujemo osnovne tehnike u oblasti o kojoj govorimo. Ovoga puta to je konščenje grafike u *Windows* programima. Može se postaviti pitanje o kakvom programiranju grafike je reč, kada su *Windows* potpuno grafički nastrojani, i kada sve ono što smo prikazali u prošlim brojevima već proizvodi neku vrstu grafike?

Sve izlazne rutine koje smo dosad konstruisali uglavnom rutine konsničkog interfejsa složene su od grafičkih elemenata. To su, ipak, rutine višeg nivoa i programer se pri njihovom korišćenju ne oseća previše grafikom koja je u njih ugrađena. Predmet našeg interesovanja je neposredno korišćenje elementarnih grafičkih postupaka – ispisivanje proizvoljnog prikaza u radni prostor prozora.

SA TEORIJE...

Počnimo od objašnjenja termina koji se koriste u primerima i literaturi.

Graphic Device Interface je interfejs između uređaja koji se koristi za prikaz grafike i aplikacije. Ranije smo već naveli da je karakteristika *Windows* okruženja nezavisnost aplikacija od uređaja. Skup rutina koje predstavljaju GDI brine upravo o konverziji onoga što aplikacija šalje u spoljni svet u formu vidljivu na odgovarajućem uređaju (ekran, štampač, ploter).

Display Context je interna struktura podataka o karakteristikama izlaznog uređaja i aktivnom prozoru. Ovi podaci se koriste za svaki grafički ispis.

GDI tools – osnovni alati za formiranje grafičkog izlaza: *pens* (pero tipovi linija), *brushes* („premazi" za popunjavanje površina), fontovi.

Bitmaps su u konkretnu GDI snimci već formiranih slika. Nije moguće programski razlučiti od kojih je elemenata snimljena slika sa stavljena ta je manipulacija bit mapama vrlo skučena.

Mapiranje je mehanizam koji omogućava da se ispiši vrše u logičkim koordinatama. To doprinosi nezavisnosti aplikacije a otvara i prostor za neka „elegantna" programska rešenja.

Izlazne rutine se neposredno koriste za crtanje, ispisivanje teksta i manipulaciju bit mapama. Odlikuje ih relativno nizak nivo složenosti u odnosu na druge komponente *Windows* okruženja.



Osnovni grafički postupak bio je pnsutan i u prethodnim primerima: to je **Paint** funkcija kojom svaki prozor-program obrađuje primljenu WM_PAINT poruku. Obrada se obično sastoji iz kreiranja *display context*-a, ispisa sadržaja i javljanja sistemu da je posao završen. Ova funkcija uglavnom zadovoljava potrebu za ispisivanjem inicijalnog sadržaja svakog prozora i kasnijeg osvežavanja narušenog sadržaja. Za složenije zadatke potrebno je ispis izvršavati i kao reakciju na neke druge poruke. Često je to potrebno izvesti kao rezultat pomeranja miša. Rekli smo već da sistemske radnje mišem ne zahtevaju posebnu obradu u programu, pokretanje miša i sa njim kursora, do opcija u meniju, klik na opciju: sistem će već prevesti u odgovarajuću poruku o izboru opcije u meniju – za program je transparentno da li je to uradeno mišem ili korišćenjem tastature.

Postoji i mogućnost mnogo detaljnija kontrole kretanja miša: sistem šalje poruku WM_MOUSEMOVE za svako pomeranje miša, poruku WM_LBUTTONDOWN kad je pritisnuto levo dugme na mišu, WM_LBUTTONUP kad je dugme otpušteno i još nekoliko različitih tipova poruka koje su vezane za upotrebu ove sprave. Uz svaku od ovih poruka prosleduju se koordinate kursora u trenutku kad je poruka formirana.

Jedan od aspekata konščenja grafike je i upotreba boja (ovo kao i mnoge druge stvari iz ove oblasti nije moglo da stane u jedan primer programa). Princip kojim se kodiraju odje u *Windows* okruženju je prilično široko i dalekovido zamišljen. Svaka odja se kodira po učešću crvene, zelene i plave u njoj (poznati RGB princip). Broj koji predstavlja boju je 24-bitni po 8 bita za intenzitet svake od tri osnovne boje. Teorijski to otvara mogućnost za konščenje 16 miliona odja, ali naravno od sposobnosti hardvera zavisi koliko će ih se stvarno raspoznavati u prikazu. U svakom slučaju, na ugrađenim drajverima je da odluče kako će to predstavljati na sistemima sa 2, 16, 256 boja.

Crtanje i odjenja u programima za *Windows* namece potrebu da se razresi još jedan problem: odnos između onog što se crta i pozadine. Kod složenijih prikaza, delovi crteza se nanosa jedni preko drugih. Pri tome može doći do „utapanja" delova crteža u pozadinu. Kontrole odnosa između delova prikaza se ostvaruje preko ROP kodova (*Raster Operation Codes*) i modalitela pozadine (*background modes*). Postoje dve vrste ROP kodova jedni odreduju odnos između linija pera (pen) i pozadine, a drugi između bitmapa koje se „nanose" i pozadine. Tako se kodom R2_NOT sprečava da crtanje odicnim (crnim) perom na crnim delovima pozadine ostane nevidljivo, jer će ovaj kod učiniti da pero ostavlja trag bojom inverznom boji pozadine.

Ulogu modalitela pozadine najlakše se može primeniti kod ispisa teksta na obojenu pozadinu. Ako prazan prostor u slovima i oko njih lakode prekriva pozadinu, onda je to „OPAQUE". Ispis. Ako se kroz slova vidi originalana pozadina, ispis je „TRANSPARENT".

...NA PRAKSU

Pokušajmo sad da kroz primer sagledamo primenu osnovnih elemenata. Program koji opisujemo je jednostavna igra na ekranu se pojavljuju i nestaju mete koje treba slici krstastim kurzorom i „pogoditi" prilskom na dugme.

BumPaint funkcija (listing 2) u našem primeru je jednostavna: iscrtava metu na tekućim koordinatama i ispisuje trenutni rezultat igra. Za crtanja mete posloji više mogucnosti, a mi smo se opredelili za jednu od jednostavnijih. Koristeći pomocni program *SDKPaint* nacrtali smo ikonu odgovarajuceg oblika. Ikona je bitmapirani crtež snimljen u odredenom formatu. Formiranu ikonu naveli smo u opisu programskih resursa (listing 3 – bum.rc) te je pri kompilaciji i linkovanju ona uključena u izvršnu verziju programa (bum.exe). Da bi se ikonom mogla manipulisati u programu potrebno je negde u inicijalnom delu C koda pozvati funkciju **LoadIcon** (listing 1 – **MainWndProc** funkcija). Ova funkcija vraća oznaku učitane ikone (**icon handle**) koja se koristi za dalji rad. Tako se nase crtanje ikone sada svodi na jedan poziv funkcije **DrawIcon** sa oznakom ikone i koordinatama za njeno prikazivanje. Kao i druge izlazne grafičke funkcije i ova kao jedan od argumenata ima oznaku *display context*-a, na kome će izlaz biti prikazan.

Funkcija za obradu poruka – **MainWndProc** sadrži za nas nove elemente. Poruka tipa WM_CREATE nastaje u trenutku kada je prozor upravo kreiran i znaci otprilike „sad može ostala inicijalizacija". Poruku šalje sistemska funkcija **CreateWindow** neposredno pre vracanja kontrole (pre **return** instrukcije). Inicijalizacija koju nas program vrsi kao odgovor na ovu poruku, pored učitavanja ikone u memoriju, jeste i aktiviranje sistemske štoperice (*timer*).

Aktiviranje štoperice funkcijom **SetTimer** nudi dva mogućnosti: da se u zadatom intervalima poziva funkcija čija se adresa zada, ili da se umesto ove adrese prosledi vrednosi NULL, nakon čega će štoperica u zadatim intervalima prozoru predevati po jednu poruku tipa WM_TIMER. Moguća je uključiti više nezavisnih štoperica. Štoperica smo upotrebili da u pravilnim vremenskim razmacima pomeramo



Listing 1

```
/*-------------------------------------------------------------------------*/
/*  Obrađuje sve poruke koje stižu aplikaciji                              */
/*-------------------------------------------------------------------------*/
long FAR PASCAL MainWndProc(hWnd, message, wParam, lParam)
HWND hWnd;
unsigned message;
WORD wParam;
LONG lParam;
{
    FARPROC     lpProcAbout;

    switch (message) {
        case WM_COMMAND:
            switch (wParam) {
                case IDM_ABOUT:
                    lpProcAbout = MakeProcInstance(About, hInst);
                    DialogBox(hInst, "AboutBox", hWnd, lpProcAbout);
                    FreeProcInstance(lpProcAbout);
                    break;

                case IDM_EXIT:
                    DestroyWindow(hWnd);
                    break;

                case IDM_LVLINC:
                    if (DELAY < MIN_DELAY)
                        break;
                    DELAY /= 2;
                    KillTimer (hWnd, 1);
                    SetTimer (hWnd, 1, DELAY, (FARPROC) NULL);
                    break;
                case IDM_LVLDEC:
                    if (DELAY > MAX_DELAY)
                        break;
                    DELAY *= 2;
                    KillTimer (hWnd, 1);
                    SetTimer (hWnd, 1, DELAY, (FARPROC) NULL);
                    break;

            }
            break;

        case WM_CREATE:
            hbmIcon = LoadIcon(hInst, "bum");
            SetTimer (hWnd, 1, DELAY, (FARPROC) NULL);
            break;

        case WM_PAINT:
            BeginPaint(hWnd, (LPPAINTSTRUCT)&ps);
            EndPaint(hWnd, ps.hdc);
            BeginPaint(hWnd, (LPPAINTSTRUCT)&ps);
            break;

        case WM_LBUTTONDOWN:
            DumTM(hWnd, lParam);
            break;

        case WM_SIZE:
            InvalidateRect(hWnd, NULL, TRUE);
            break;

        case WM_TIMER:
            DrawNewTarget (hWnd);
            break;
        case WM_DESTROY:
            DeleteObject(hbmIcon);
            KillTimer (hWnd, 1);
            PostQuitMessage(0);
            break;

        default:
            return (DefWindowProc(hWnd, message, wParam, lParam));
    }
    return (NULL);
}
```

Obrada svih poruka koje stižu aplikaciji

Listing 2

```
/*-------------------------------------------------------------------------*/
/*   Crta metu i ispisuje rezultat                                         */
/*-------------------------------------------------------------------------*/
void RunPaint (hWnd, oDC)
HWND  hWnd;
HDC   hDC;
{
    Rect    Rect;

    DrawIcon (oDC, (short)TX, (short)TY, hbmIcon);
    sprintf (sm, " Score % 5d   Delay % 5d                    % 5d",
score, DELAY, stmp);
    GetClientRect(hWnd  (LPRECT) &Rect);
    TextOut (hDC,
             (short) 10,
             (short) Rect.bottom-15
             (LPSTR) ws,
             (short) strlen(ws));
}
```

Crtanje mete i ispisivanje rezultata

Listing 3

```
#include "windows.h"
#include "bum.h"

bum ICON bum.ico

BumMenu MENU
BEGIN
    POPUP       "&Options"
    BEGIN
        MENUITEM    "Speed+",               IDM_LVLINC
        MENUITEM    "Speed-",               IDM_LVLDEC
        MENUITEM    SEPARATOR
        MENUITEM    "Pexit"                 IDM_EXIT
        MENUITEM    SEPARATOR
        MENUITEM    "&About Bum ..."        IDM_ABOUT
    END
END

BumAcc ACCELERATORS
BEGIN
    0x41,       IDM_ABOUT,      VIRTKEY, ALT
END

AboutBox DIALOG 22, 17, 144, 75
STYLE DS_MODALFRAME | WS_CAPTION | WS_SYSMENU
CAPTION "About Bum"
BEGIN
    CTEXT "Bum Application"     -1,       0, 10, 144,  8
    CTEXT "Version 1.0"         -1,       0, 26, 144,  8
    CTEXT "(c)Mumuri 75/91"     -1,       0, 40, 144,  8
    DEFPUSHBUTTON "OK"          IDOK,    55, 59,  32, 14,     WS_GROUP
END
```

Risors detoteke

Listing 4

```
/*-------------------------------------------------------------------------*/
/*   Brisu prethodnu metu i crta na novoj poziciji                         */
/*-------------------------------------------------------------------------*/
void DrawNewTarget (hWnd)
HWND  hWnd;
{
    HDC     oDC;
    Rect    Rect;

    stmp++;
    oDC = GetDC (hWnd);
    GetClientRect(hWnd, (LPRECT) &Rect);

    /* brisanje prethodne mete    */
    BitBlt (hDC, (short)TX, (short)TY, 25, 25, NULL, 0, 0  WHITENESS);

    /* crtanje sledece mete       */
    TX = (long)rand() * (long)(Rect.right-50) / (long)RAND_MAX;
    TY = (long)rand() * (long)(Rect.bottom-50) / (long)RAND_MAX
    DrawIcon (hDC, (short)TX, (short)TY  hbmIcon);

    /* ispis rezultata            */
    sprintf (ws, " Score % 5d   Delay % 5d                    % 5d",
score, DELAY, stmp);
    TextOut (hDC
             (short) 10,
             (short) Rect.bottom-15,
             (LPSTR) ws
             (short) strlen(ws));
    ReleaseDC (hWnd  hDC);
}

/*-------------------------------------------------------------------------*/
/*   Ispituje pogodak                                                      */
/*-------------------------------------------------------------------------*/
void Bum (hWnd, lParam)
HWND  hWnd;
long  lParam;
{
    HDC         hDC;
    Rect    Rect;
    POINT       mp;

    mp = MAKEPOINT(lParam);

    if (mp.x >= TX && mp.x < TX+25 && mp.y >= TY && mp.y < TY+25)
    {
        score++;
        hDC = GetDC (hWnd);
        GetClientRect(hWnd, (LPRECT) &Rect);
        sprintf (ws, " Score # 5d   Delay % 5d                    %.5d",
score, DELAY, stmp);
        TextOut (hDC
                 (short) 10,
                 (short) Rect.bottom-15
                 (LPSTR) ws
                 (short) strlen(ws));
        ReleaseDC (hWnd  hDC);

        SetVoiceNote (1, 20, 1, 1);
        StartSound ();
        WaitSoundState (S_ALLTHRESHOLD);
        StopSound ();
    }
}
```

Brisanje prethodne mete i crtanje na novoj poziciji

metu po ekranu. Ze zaustavljanje stoperice koristi se funkcija **KillTimer**.

Poruku WM_TIMER obrađujemo funkcijom **DrawNewTarget** (listing 4). Obavezno je povezivanje sa **display context**-om pre samog crtanja. Napomenućemo samo da je funkcija **BitBlt** koju ovde koristimo ze brisanje prethodne mete jedna od najčešće korišćenih u *Windows* sistemu. Njena osnovna namena je kopiranje svih vista bitmapa bez obzira da li se one nalaze u istom *display context* u. ili su čak pridružene različitim uređajima. Ostatak funkcije **DrawNewTarget** predstavlja izračunavanje novih slučajnih koordinata mete i crtanje ikone na novoj poziciji.

"Nišanisku spravu" smo u igri obezbedili korišćenjem jednog od predefinisanih tipova kurzora – IDC_CROSS. Kurzor je sistemski vezan za kretenje miša, tako da nama ostaje samo da registrujemo pritisak na dugme. Poruka WM_LBUTTONDOWN kao jedan od parametara (lParam) nosi i koordinate kurzora u okviru aplikacijskog prozora u trenutku pritiska na dugme. Kroz funkciju **Bum** (listing 4) ispituje se poklapanje položaja kurzora sa položajem mete.

Ovako jednostavnim primerom samo je načeta paleta grafičkih mogucnosti *Windows* sistema, ali je program vrlo dobro potvrdio tezu o efikasnosti koda sa početka senje. Invijalni primer iz uvodnog teksta, od oko 250 redova C programa porastao je za samo 50 redova, a dobili smo malu *Windows* igru!

Kompletan izvorni i izvršni kod ovog programa možete naći na Sezamu.

PREVIEW U KLIPERU

Potrebno je odštampati neki izveštaj u *Clipper*-u? Ništa lakše: LIST ALL TO PRINTER. I još samo stotinak linija programa pre toga za izbor raznih opcija štampanja. I tako stotinak puta.

Nenad Batoćanin

Ako uporedite tipičan *Clipper* program sa programom pisanim na nekom standardnom programskom jeziku videćete bitnu razliku u broju ulazno-izlaznih podataka obično *Clipper* program ima mnogo više i ulaznih i izlaznih. Naravno, ovo nema mnogo veze sa jezikom – u pitanju je **tip problema** čijem je rešavanju jezik namenjen. Pošto je *Clipper* namenjen rešavanju problema sa velikim brojem podataka, naredbe koje se odnose na ulaz i izlaz podataka veoma su značajan element jezika. Jedan deo tih naredbi čine funkcije za realizovanje komunikacija konsnik – računar preko ekrana i tastature kao na primer **AChoice**, **MENU TO** i sl. Drugi samo malo manje važan deo čine naredbe za programiranje periferijih uređaja – u prvom redu štampača. Kada se uzme u obzir da se u oblasti poslovne obrade podataka kao zvaničan dokument priznaje jedino tekst na papiru, jasno je kolika je važnost štampanja dokumenata. *Clipper* za štampanje ima relativno alabašan skup mogućnosti. Mada se razne opcije naredbi odnose na direktan izlaz na štampač, upotrebljivost takvih naredbi prilično je simbolična i one su primenljive samo u krajnje jednostavnim slučajevima. Na svu sreću *Clipper* ima mogućnosti da radi sa datotekama i štampačem na dosta „niskom" nivou, postoje nje takvih funkcija nam omogućuje da razvijamo sopstvene procedure za štampanje izveštaja.

Pošto je pisanje izveštaja povezano sa masom čisto tehničkih detalja (promena kvaliteta štampe, izbor numeracije strana), cilj nam je da razvijamo sistem koji će maksimalno rasteretiti programera brige oko tehničkih detalja i omogućiti mu da pažnju usmeri na program koji generiše izveštaj, a da konsnik uvek može na jednostavan način da izabere opcije koja mu u datom trenutku odgovaraju.

DIREKTNO ...

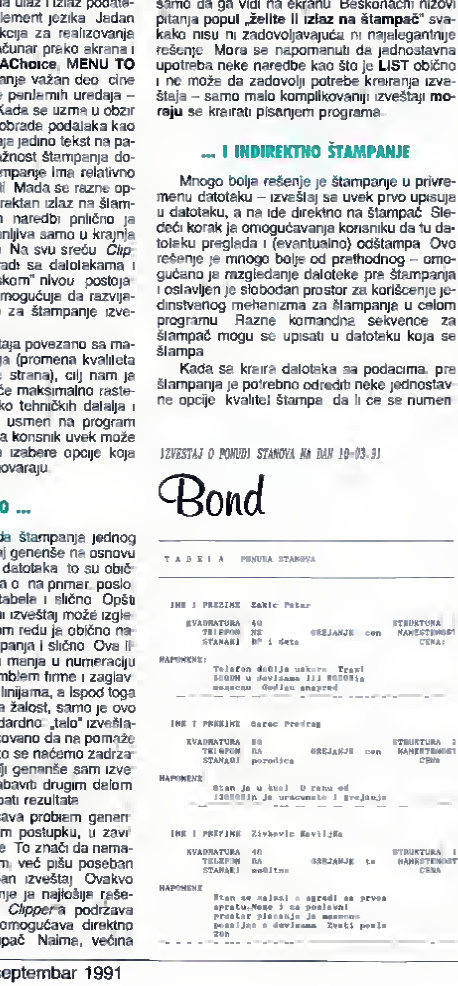
Kako uopšteno izgleda štampanje jednog izveštaja? Prvo se izveštaj generiše na osnovu podataka iz jedne ili više datoteka (to su obično naki dugi nizovi brojeva o nečemu, na primer, poslovanju po danima, neke tabele i slično). Opšti obrazac za neki zamišljeni izveštaj može izgledati kao na slici 1. U prvom redu je obično naziv izveštaja, datum štampanja i slično. Ova linija se na sledeću stranu manja u numeraciju strane. Ispod se nalazi amblem firme i zaglavlje dokumenta označeno linijama, a ispod toga su neki nizovi brojeva. Na žalost, samo je ovo zaglavlje manja-više standardno „talo" izveštaja je obično toliko komplikovano da na pomaže nikakav automatizam. Zato se nećemo zadržavati na delu programa koji generiše sam izveštaj. Sada ćemo se pozabaviti drugim delom kako pregledati i odštampati rezultate.

Dosta programera rešava problem generisanja izveštaja po kratkom postupku, u zavisnosti od ukazana situacije. To znači da nemaju nikakav razrađen sistem, već pišu poseban program za svaki potreban izveštaj. Ovakvo (obično direktno) štampanje ja najlošije rešenje, iako dosta komandi *Clipper*-a podržava opciju **TO PRINT**, koja omogućava direktno slanje rezultata na štampač. Naime, većina ovih komandi ne omogućava nikakvu preciznu kontrolu štampača, počev od prestanka štampanja pre stvarnog kraja (kad korisnik pritisne određeni taster), pa do raznih postavljanja fontova i kvalitete ispisa. Vrlo često konsnik i ne želi stvarno da odštampa izveštaj, već želi samo da ga vidi na ekranu. Beskonačni nizovi pitanja popul „**želite li izlaz na štampač**" svakako nisu ni zadovoljavajuća ni najelegantnije rešenje. Mora se napomenuti da jednostavna upotreba neke naredbe kao što je **LIST** obično i ne može da zadovolji potrebe kreiranja izveštaja – samo malo komplikovaniji izveštaji **moraju** se kreirati pisanjem programa.

... I INDIREKTNO ŠTAMPANJE

Mnogo bolje rešenje je štampanje u privremenu datoteku – izveštaj se uvek prvo upisuje u datoteku, a na ide direktno na štampač. Sledeći korak je omogućavanja korisniku da tu datoteku pregleda i (eventualno) odštampa. Ovo rešenje je mnogo bolje od prethodnog – omogućano je razgledanje daloteke pre štampanja i oslavljen je slobodan prostor za korišćenje jedinstvenog mehanizma za štampanja u celom programu. Razne komandna sekvence za štampač mogu se upisati u datoteku koja se štampa.

Kada se kreira daloteka sa podacima, pre štampanja je potrebno odrediti neke jednostavne opcije: kvalitet štampe da li će se numeri-



sati stranice, itd. Metod „postavljanja pitanja" za razne opcije pri štampanju ne dolazi u obzir ako želite da iznervirate korisnika, postoje i mnogo humanji načini! Sa druge strane, kad bi programer kod svakog izveštaja posebno predvidao šta sve može da zatreba konsniku, sigurno je da ne bi bio previše produktivan. Cilj nam je da razvijemo sistem koji će, sa jedne strane, omogućiti da lako programirate i najkompilikovanije izveštaje, a sa druge strane da korisnik uvek i uz malo pritisnutih tastera može jednostavno da razgleda izveštaj, da ga štampa i da bira pri tom razne opcije.

PREGLED IZVEŠTAJA

Datoteka sa izveštajem lako se kreira lako šlo se izlaz programa preusmeri u datoteku komandama **SET ALTERNATE** ili **SET PRINTER TO**. Problem koji je našlo teže rešiti je pregled tako kreirana datoteka. Kao uzor nam mogu posluziti mnogi programi tipa **LIST** ili **README** – to su mali programi veličina nekoliko kilobajta koji omogućavaju komotno razgledanje teksta. Nezanemarljiv broj programera i konsti neki od ovih programa za pregled datoteka pre štampanja. Ako se i zanamari zabrana korišćenja ovih programa u takve svrhe, mislim da je korišćenje komande **RUN** maksimalno knvljenje konstrukcije programa – tome treba pribegavati samo u krajnjoj nuždi.

MemoEdit funkcija je prvi kandidat koji liči na ove programe – omogućava udobno razgledanje teksta i pri tome možemo da je proizvoljno konfigunšemo konsničkim funkcijom. Međutim, jedan važan detalj čini u ovom slučaju funkciju **MemoEdit** neupotrebljivom: maksimalna dužina teksta koji može učitati **MemoEdit** je 64K, a to je daleko ispod veličine koju može imati jedan izveštaj. Na primer, 10000 slogova po 100 bajtova čini datoteku od oko 1MB. Možda bi se moglo izvesli da **MemoEdit** učitava tekst u blokovima, ali imam utisak da bi to bilo „skrpljeno" rešenje.

Interesantna je ideja da se za pregled teksta upotredi **DBedit** funkcija običnu takst datoteku možemo konvertovati u **DBF** datoteku sa slogom **CHARACTER 80**, čime smo rešili problem dužina daloteke. Međutim, sad se javljaju novi problemi: šta sa datotekom koja ima red šin od 80 kolona?

Rešenje koje je najprihvatljivije je pisanje programa koji omogućava listanje daloteke proizvoljne veličina. To se može uraditi i u *Clipper*-u, ali je previše sporo. Ostaju nam *assembler* i *C*.

U algoritmu je principijalno sve jasno, ali realizacija nameće neke praktične probleme. Prvi problem je koncepfualne prirode. Naime, moramo se odlučiti za neki odnos memorija/brzina. Šlo je naš program brži, potrošićemo više memorije i obrnuto. Ovo univerzalno pravilo o održanju memorije možemo ovako pojasniti. Čitanja podataka sa diska je najsporija operacija. Dakle, ako želimo najbrži mogući program, učitaćemo prvo **ceo** takst u memoriju i tamo ga obradivati daleko brzim funkcijama. Samo kako na taj način obraditi datoteku od nekoliko megabajta? Kad se uzme u obzir da taj program treba smestiti u memoriju zajedno

Slika 1

```
IZVESTAJ O POSLOVANJU  02.05.91
  ANALIZA

NASLOVNJE IZVESTAJA

NASLOV 1      NASLOV 2        NASLOV 3

<pod 1>       <pod  >         <pod  >
<pod 2>       <pod  >         <pod  >
<pod 3>       <pod  >         <pod  >
```

Opšti primer štampanog izveštaja

Slika 3 GetLine i ADown

```
;*************************************************************
; DATUM: 24  7 1991
;
; NASLOV: GetLine
;
; OPIS:   Ucitava liniju broj AX; BX=File Handle
;         Ucitavanje polazi od TxtEnd pozicije u datoteci
;         ako je na iskaru C, Nesto je do kraja datoteke
;*************************************************************

GetLine PROC NEAR

   PUSH AX
   PUSH AX                      ; Sacuvaj broj linija

   MOV AX  4200h                ; Pomeranje pokazivaca na TxtEnd
   MOV CX  TxtEndR              ; TxtEnd => CXdx
   MOV DX  TxtEndL
   INT 21h                      ; File Seek

   POP AX                       ; Vrati broj linija
   MOV CX  AX
   MOV Ge  AX
   DEC Ge                       ; Numeracija ide od 0

   MOV BX  OFFSET Screen        ; Ge:DX -> Mem  Buffer
   dec CX                       |
   JS ReadL                     | Ucitava se prva linija

LINe:                           ; Odredi adresu poc. linije
   ADD Ba  LineMax              | Sledeca linija
   LOOP LINe

ReadL:
   MOV DI  DX                   ; Za kasnije

   MOV AH  3Fh                  | Read File
   MOV CX  LineMax              | Cita se cela linija
   INT 21h                      | Citaj
   OR AX, AX                    | Da li je nesto procitano?
   JE EofXL                     | Ceo||

   MOV BX, AX                   ; Sacuvaj ucitanu duzinu linija

   PUSH Ge
   POP Ba                       ; Ge:DI -> pocetak linija
   MOV Ge  DI
   MOV CX  AX                   ; Trazenje <CR>
   CLD
   MOV AL  13
REPNE SCASB                     ; Trazi;
   JZ Found

   MOV AX, BX                   ; Nema <CR> - vrovatno je EOF
   MOV LineLen[Ge]  AL
   PUSH AX
   DEC AL
   JMP Clean

Found:                          ; Nevtoji <CR>
   MOV AX  DI                   ; Duzina linija
   SUB AX, DX
   INC AL
   MOV LineLen[Ge], AL
   PUSH AX
Clean:
   MOV CX  LineMax
   SUB CL, AL                   ; Koliko ja do kraja linije?
   ADD CX  2
   DEC DI                       ; Clacanja natetka linija
   MOV AL
REP STOSB

   MOV AX                       ; Duzina linija
   ADD TxtEndL  AX              ; MOVi TXT END
   MOV TxtEndR  ?               |
   CLC                          ; Bija Nuf
   JMP NEVl

XofGE:
   MOV CX  LineMax              ; Oelsti ostatu linija
   MOV AL
REP STOSB
   STC                          ; NOV indikator

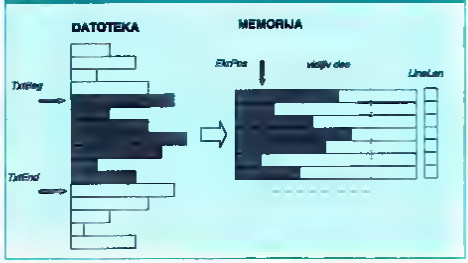
TWOL:
   POP AX
   MOV
GetLine ENDP


;**************************************************************
; DATUM: 24  7 1991
;
; NASLOV: ADown
;
; OPIS:   Pomeranje za linija dole
;         Ako se skroz se skrolaja, bipa C
;**************************************************************

ADown PROC NEAR

   MOV AX  FileBufL             ; Nuf
   SUB AX  TxtEndL
   MOV DX  AX
   MOV AX  FileBufG
   SBB AX  TxtEndR
   OR AX, Ge
   JA COMP
```

Slika 2



Datoteka koja se lista i učitavanje njenog dela u memorijski bafer

```
   JMP FinishScr

ComAD:
   ADD AX  AN                   ; TxtBeg = TxtBeg + Len [hed]
   MOV AL, HeaderLen
   MOV Ge, AX
   MOV AL  LineLen[BP]
   MOV TxtBegL  AX
   MOV TxtBegH  0

   PUSH DS                      ; Skrol Bafera
   POP ES                       ; Ge == ES = FGG (BASE)
   LEA DI, LineLen[BP]
   MOV SI, DI
   INC SI                       ; |SI| --> |DI|
   MOV CL  PageLen
   XOR CH, CH
REP MOVSW

   MOV DI, BufEnd               ; Na:DX -> nasak
   MOV DI, DI                   |
   ADD SI  LineMax              ; Ge:SI -> na sledeci red
   MOV CX  LineMax*LineMax      ; Koliko treba pomeritit
   SUB CX  DI                   |
   SUB CX  OFFSET Screen        |
   CLD                          |
REP MOVSB                       ; Pomeri

Ge:
   MOV AX  24                   ; Dumi sadnja linija
   CALL  GetLine                |
   JNC Finish                   ; Nija EOF

   CLC
   JMP Finish

FinishScr:
   STC                          ; Nova lin  nija prihvacena

Finish:
   NOV
ADown ENDP
```

Procedure GETLINE i ADOWN

Slika 4 Attr

```
; --- Tabela video atributa
NormAttr    Ge 07h              ; Normal atribut    (0)
UndAttr     DB 01H              ; Normal Underline  (1)
BoldAttr    DB 0fh              ; Bold              (2)
BoldUnAttr  de 09H              ; Bold Underline    (3)
InvAttr     Ge 70h              ; Zavarsno atribut  (4)
InvUnAttr   Db 71h              ;                   (5)
InvBoldAttr DB 79h              ;                   (7)
InvUnAttr   Ge 71h              ;                   (8)

ColorTable  MyU NormAttr        ; Pocetak tabele atributa

; --- Nemale kadeva atributa u datoteci
UpIge       MoV 1               ; Uklucuvanje Underline
BPlus       MoV 2               ; Ukljucuje Bold
IPlus       MoV 4               ; Ukljucuje Italic

UNless      MyU 5               ; Iskljucuje Underline
BNless      MyU 8               ; Iskljucuje Bold
INless      MoV 8               ; Iskljucuje Italic

Norm        MoV 9               ; Iskljucivanje svih atrib

;**************************************************************
; DATUM: 22  7 1991
;
; NASLOV: DispLine
;
; OPIS:   Ispisuje liniju broj CX
;**************************************************************

DispLine PROC NEAR

   PUSH Ge
   MOV Ge  BPFOET ColorTable ; Ge -> ColorTable
   PUSC CX
   MOV AX  CL                ; AX = Broj reda

   MOV BG, VAddr             ; Ge:DI -> Video Mem
   MOV DI  0

   MOV BX  OFFSET Screen     ; Na:SI -> NASFer
   DEC CX
   JS Disp

LineNo:                      ; Odredi poc  linija
   Mov Bl  LineMax           ; Sledeca linija
   ADD Gl, 160               | NimBoli+atributi
   LOOP LineNo
```

Listing se nastavlja na sledećoj strani

Slika 4 Nastavak sa prethodne strane

```
Disp:
   MOV AL, 4                    ; INVERT
   CMP AH, DoesNotLes           ; Da li treba invert?
   JLE Sym00                    ; Ukljucivanje ili iskljucivanje
   XOR AL, AL                   ; NORMAL
Sym00:
   MOV CL, 0                    ; Broj simbol koji se stampa?

L1:
   MOV AH, [BX]                 ; Uzmi simbol
   CMP AH, 10                   ; Ako je kod<10 to je kontrolni
   JB NoCtrl
   CALL Attr                    ; Analiticki atribut
   INC BX
   JMP L1
NoCtrl:
   CMP CL, EkrPos               ; CL = EkrPos ?
   JE EqW
   INC BX
   INC CL
   JMP L1

EqW:
   MOV CL, 80                   ; Kopira se 80 simbola
AgW:
   MOV AH, [BX]                 ; Uzmi simbol
   CMP AH, 32                   ; Da nije kontrolni?
   JA Move
   CALL Attr
   INC BX
   JMP AgW
Move:
   MOV BS:[BX], AH              ; Upisa simbol
   INC BX
   POPD AH
   XLAT CrtAtrTable             ; Nadji odgovarajuci atr
   MOVB BS:[BX], AL             ; Upisi atribut
   POP AH
   INC BX
   INC BX
   LOOP AgW

   POP CX
   POP BS

   RET
DispLine ENDP

;*****************************************************
; DATUM: 31  7.1991
;
; NAZIV: Attr
;
; OPIS:  Na osnovu kontrolnog simbola u AH,
;        nasja atribut u AL
;*****************************************************

Attr PROC NEAR

   CMP AH, IPlus                ; Da li je PLUS?
   JA NoPlus
   OR AL, AH                    ; Setaj odgovarajuci atribut
   RET

NoPlus:
   CMP AH, IMinus               ; Da li je MINUS?
   JA NoMinus
   SUB AH, IPlus
   SUB AH
   AND AL, AH                   ; Resetaj odgovarajuci atr
   RET

NoMinus:
   CMP AH, Norm                 ; NORMAL?
   JNE NoNorm
   XOR AL, AL                   ; Atr = 0
   RET

NoNorm:

   RET
Attr ENDP
```

Slika 5

```
#define MESEC {"januar"  "februar"  "mart"
               "April"   "maj"   "jun"   "jul"   |
               "avgust"  "septembar"  "oktobar"
               "novembar"  "decembar"}

*************************************************************
* DATUM:  4  8 1991
*
* NAZIV: Main
*
* OPIS:  Glavni program
*
*
*************************************************************

PROC Main
   SET DATE GERMAN
   SET CURSOR OFF
   // Inicijalizacija sistema stampanja
   InitPrn ()
   // Inicijalizacija ekrana
   DayScreen (Replicate("-",80*24),0,0)

   // Poziv izvestaja
   Report ()
   CLS
RETURN

*************************************************************
*
* DATUM:  4  8 1991
*
* NAZIV: Report ()
*
* OPIS:  Primer izvestaja
*
*
*************************************************************

PROC Report

   // Maksimalni broj iteracija
   Job_max := 365

   // Nova sa definisanom sirinom i naglavlje
   BenDer  = Bold(e+Ital(e+"Strana |"+Str(++Str,4)))+AttrOff
   Title := Bold(e+"IZVESTAJ o PRODAJAMA po RADNIK"+Bold(e+Bold(e+Bold(e
```

ku ponovni početak rada pokazivači se reseluju i datoteka se učitava od početka. Pomeranje na kraj je nešto drugačija. Najpravilnije rešenje je da se učita blok sa kraja datoteke da se zatim „formatira" u baferu kako bi svaka linija zauzela svoje mesto, a da se svi pokazivači poslava na odgovarajuće vrednosti. Mada ovakva analiza nije previše komplikovana može biti prilično obimna. Zato je u proceduri **EoT** koja realizuje komandu <Ctrl PgDn> primenjen drugačiji postupak. Pri izvršenju komande za skok na kraj teksta u baferu treba da se nađu poslednjih **LineNum** linija. Pošto je svaka linija kraća od neke maksimalne vrednosti **LineMax** početak dela datoteke koji treba učitati sigurno se nalazi posle **LineMax*LineNum** pozicije pre kraja datoteke. Zato je dovoljno da posle ove pozicije u datoteci načamo prvi par <CR><LF>, pa da primenimo u petlji proceduru za učitavanje zadnje linije, sve dok se na dođe do kraja datoteke. Tako će sve linije doći na pravo mesto. Pošto se analiza sada svodi samo na traženje <CR><LF> para i uzastopno pozivanje procedura **ADown**, višestruko je olakšana.

KONTROLNI KODOVI

Vrlo često je potrebno u izveštaju neke delove posebno označiti posebnim načinom štampe (odlično,). Ako pri generisanju izveštaja upišemo kontrolne kodove za štampač pojednostavićemo postupak štampanja ali cemo lako pokvariti izgled teksta pri pregledu – nezgodna osobina tih kontrolnih sekvenci je što kvare teksi pri razgledanju nije baš najprijatnije čitati nešto popul <ESC>EiZVEŠTAJ<ESC>F. Mnogo lepše rešenje je da se i pri pregledu kontrolne sekvence interpretiraju na neki način da se tekst koji će biti odštampan masnijim slovima vidi u nekoj drugoj boji i slično. Ako direktno interpretiramo sekvence za štampač moraćemo da se vežemo za jedan tip štampača, što nikako nije dobro. Osim toga, sekvence za štampač su dosta duge i nisu najpogodnije za analizu. Primenjeno je drugo rešenje. Pri generisanju izveštaja se za specijalne efekte upisuju kontrolni simboli sa kodovima ispod 10. Tako simboli sa kodom 2 označava početak podvlačenja a simboli sa kodom 7 kraj podvlačenja. Interno se u programu za svaki simbol vodi atribut. Pri tome znači imaju samo tri bita najmanje važnosti. Prvi bit (najmanje težine) označava da li je uključeno podvlačenje, drugi da li je uključena masna štampa, a treći da li je uključen kurziv (italik štampa). Na primer, atribut **0000 0101** označava početak podvlačenja sa kurzivom. Procedura **Attr** sa slike 4 određuje na osnovu učitanog kontrolnog koda u registru AH i starog atributa u AL novi atribut. U proceduri **DispLine** se ti tarni atribut pretvara u pravi atribut koji treba upisati u video memoriju na osnovu tabele svih kombinacija atributa konačnjen za tu priliku idealne instrukcije **XLAT**.

Nezgodna osobina postojanja atributa teksta je da se teksi mora **kontinualno** analizirati. Šta to znači? Na primer, ako je negde u sredini teksta kod za uključivanje podvlačenja i ako taj atribut nije do kraja teksta promenjen, onda će i kraj teksta biti „podvučen". Ako treba izvršiti komandu „skok na kraj datoteke" to se na može uradit jednostavnim preskakanjem teksta pre kraja jer se u njemu mogu nalaziti kodovi za promenu atributa koji će se odnositi na kraj teksta – zato se pri skoku na kraj datoteke mora analizirati svaki bajt batoteke od tekuće pozicije do mesta za skok. Još gora situacija je za kretanje „na gore". Na primer prilpostavite da ste upravo izvršili komandu za skok na kraj datoteke i da ste sad pritisnuli strelicu na gore. Prilhodnu liniju je lako učitati, ali šta je sa njenim atribulima? Jer možda je baš u prilhodnoj liniji (iznad te) promenjen neki atribut! Nema druge nego analizirati atribute **od početka** teksta ili ih pamtiti u nekim „prekidnim" tačkama, na primer na početku svake strane. Ovo je ujedno odgovor onima koji se pitaju zašto je skrolovanje u bilo kom editoru daleko brže nego u bilo kom tekst procesoru – teksi procesor za razliku od editora mora da analizira tekst koji prelazi.

Ovakvu analizu ne možemo prihvatiti – izveštaji mogu biti vrlo dugi (i po nekoliko megabajta), pa analiza celog teksta zbog skoka na kraj datoteke nikako nije preporučljiva. Da bismo ovo rešili bez dodatnih komplikacija, usvojićemo ograničenje da atributi važe samo do kraja reda – tada se automatski postavljaju na normalnu vrednost. Ovo nije preveliko ograničenje kad se uzma u obzir da se označavanje obično i koristi za deo linije – mnogo redi slučaj da se označava deo teksla rešava se označavanjem svake linije posebno – dodatak od nekoliko bajtova je zaista zanemarljiv.

POZIV „REZIDENTNIH" PROCEDURA

Sledeći problem koji se javlja je veza sa *Clipper*-om. Naime, treba podesiti razne parametre štampanja. Ovo se naravno može uraditi i u asembleru ali zbog raznih ukusa i raznih potreba za podesavanja mnogo je odije ovo raditi u *Clipper*-u kako bi se omogućile lakše izmene. Zatim, oam jednostavnog izlaza iz procedure **View** (na primer zbog štampanja) potrebno je predvideti i pozive procedura koje su dodaljene nekim lasterima, na primer, kalkulatora za računanje i slično. *Clipper* interno vodi evidenciju pritisnutih tastera, tako da rutine **KEY** tipa ne važe vari njega. Naravno najlakše je uvesti zabranu upotreba rezidentnih procedura, ali verovatno su one i najpotrebnije pri razgledanju raznih izveštaja. Značl, poslavlja se problem kako iz asemblera pozvati *Clipper* proceduru. Ovaj problem se dosta često javlja i nije ga uvek jednostavno rešiti. Dokumentacija o ovome mudro cuti, a analiza *Clipper* **OBJ** datoteke ne daje neka jasne obaveštenja o tome – očigledno se radi o nekom nestandardnom postupku.

Najjednostavnije bi bilo da se napravi tabela svih tastera posle čijeg pritiska se prekida procedura **View** i vraća kontrola *Clipper*-u. Posle toga bi nadređena *Clipper* procedura pozvala odgovarajuću proceduru (na primer, kalkulator) i zatim ponovo pozvala proceduru **View** ili nastavila dalju obradu. Međutim, smeta jedna neprijatna sitnica: procedura **View** će teksi listati uvek od pocetka, to znači da ako na sredini teksta pozovete kalkulator, posle završetka računanja, tekst će se pomeriti na početak, a to sigurno ne želite. Znači, rešenje je u **privremenom** napuštanju procedure **View** – po povratku se listanje nastavlja gde je prekinuto. Da bi *Clipper* procedura znala koji je taster pritisnul, procedura **View** je napravljena kao funkcija, vraca simbol koji označava koji je taster pritisnul. Tako „Q" označava da je pritisnul taster <Esc> i da treba prekinuti listanje, a „C" označava da je pritisnuta kombinacija <Alt-k> što može označavati poziv kalkulatora; i tada se posle poziva kalkulatora kontrola ponovo vraća funkciji **View**. Stvar se dosta jednostavno rešava samo uvođenjem indikatora nastavka **ContInd**: oslali parametri se ponašaju kao **STATIC** promenljiva i vrednosi im ostaje sačuvana po ponovnom pozivu procedure **View**. Na pritisak kojih tastera treba privremeno napustiti program, određuje se pretraživanjem tabele **TastTable** sastavljene od parova (<kod tastera>, <signalni simbol>).

PROGRAMIRANJE IZVEŠTAJA

Najvažniji deo je onaj koji čete svakodnevno upotrebljavati – deo za pisanje izveštaja. Ovaj deo je maksimalno pojednostavljen kako bi se pažnja što više usmerila na sam izveštaj. Pre korišćenja sistema, na početku glavnog programa treba pozvati proceduru **IniPtr** koja inicijalizuje ceo sistem. Primer programa koji generiše izveštaj dat je na slici 5. Sva se radi uz pomoć samo tri procedura: **Prt1**, **Prt2** i **Prt3**. Posle **Prt1** treba odštampati zaglavlje izveštaja, a posle **Prt2** telo izveštaja. Pri pisanju izveštaja štampanjem globalnih promenljivih **BoldON**, **BoldOFF**, **UndON**, **UndOFF** itd. menjate alnbuta teksta. Može se promeniti zaglavlje strane (globalna promenljiva **Header**) i naslov izveštaja (globalna promenljiva **Title**). Osim normalnih kontrolnih simbola u zaglavlju alrane se može upisati i specijalni simbol \# koji će se pre štampanja zameniti brojem teku će strane. Strana se prekida na mestu kada broj linija pređe maksimalan dozvoljen broj ili kada u izveštaju odštampate kontrolni string **NewPage**.

Za vreme generisanja izveštaja, naročito ako duže traje, zgodno je da korisnik zna koliko će još obrada trajati. Zato se uvek ispisuje procenat urađenog posla, da bi to bilo korektno za svaki izveštaj, pre generisanja morate odrediti približnu maksimalnu vrednosi brojača posla. Za primer obrade svakog sloga neke datoteke, ta vrednosi iznosi **LastRec()**. Posle loga svako pozivanje funkcije **Job (n)** uvećava brojač posla za **N**, a na pokazivaču posla se ispisuje procenat urađenog posla.

Kako rade ove tri procedure? **Prt1** usmerava izlaz u datoteku **PART1**, **PRT** koja sadrži zaglavlje izveštaja. **Prt2** usmerava izlaz u datoteku **PART2.PRT**, a **Prt3** realizuje razgledanje izveštaja i (eventualno) štampanje. Svaki kontrolni kod se pre štampanja linije zamenjuje odgovarajućim kodom štampača.

Pre početka izvešlaja štampa se zaglavlje (obično logo firme i sl.). To je običan ASCII teksl u baloteci **AMBLEM.TXT** koji se može proizvoljno menjati. Umesto ovoga možete ubaciti i štampanje zaglavlja u grafičkom modu (videti prilhodne brojeve *"Računara"*).

ŠTA NEDOSTAJE

Naravno, ovakav sistem ima i nedostataka. Neke stvari, kao na primer preciznija kontrola štampača i veći izodr kontrolnih kodova jednostavno se dodaju izmenom procedura **Prt3** i **GoPrint**. Mnogo osetljivije je samo štampanje. Naime, ako štampač nije dobro povezan funkcijom **IsPrintar** se pre početka štampanja provarava da li se štampanje može otpoceti. To nas oslobađa greške lipa zaodravljen prilsak na laster *ON LINE*. Međutim, ako u toku rada nastane greška, stvar je mnogo nezgodnija. Donekle se problem ublažava ako se pre svakog štampanja linije proveravaa da li je štampač spreman za rad, ali greška se može desiti i u sredini. Tada se problem jedino može rešiti končenjem *Clipper Error* sistema koji omogućuje potpunu kontrolu grešaka.

Jedno moguće proširenje sistema se može odnositi na samo genenisanja izveštaja: opis izveštaja se može držati u posebnoj datoteci a štampanje bi se moglo izvesti nekom procedurom tipa **REPORT FORM**.

Sva procedure možete naći na „Sezamu" – nisu objavljene zbog duzine. Pri izradi programa je korišćen *Clipper* verzija 5.01 ali polpuno ista procedura (osim onih PRG tipa) važe i za verziju *Summer 87*.

DINAMIČKI OBJEKTI

Ova serija tekstova se bavi glavnim osobinama jezika C++. Veliki deo jezika C++ poklapa se sa jezikom C i te, zajedničke, osobine ovde necemo posebno objašnjavati. C++ uvodi niz novina u odnosu na sam C i uzdiže ga na nivo objektnog jezika. Skoro svaki C program će se prevesti i pod bilo kojim C++ sistemom, ali obratno ne važi. Ovaj nastavak, ugiavnom, posvećujemo C dinamici

Duško Savić

U prošlom nastavku nismo do kraja objasnili preoplarecivanje funkci ja U primeru na list ngu **a34 cpp** imena triju funkcija **preopFunk** bila bi prevedena u nešto popul

```
_preopFunk
_preopFunK_
_preopFunKl_l_
```

ili slično Programer to normalno ne vidi, mada mu svaki bolji razvojni sistem za C++ takvu mogucnost slavlja na raspolaganje obično raznim skralnicama tipa +**g -g** Linkera što se lice nikakvog konllikle nema – sva imena su različita Obratimo pažnju da prvo ime nije unakazeno već ga linkar vidi kao ime obične C funkcije (takvoj funkcji se standar dno dodaje ortica _ ispred imena) Zato C++ može da saraduje sa funkcijama iz običnog C a U komplikovanijim slučajevima Taj dizajn je opasen ako se deklaracije pomešaju npr ako imamo u programu obič nu C funkciju **preopFunk** i još dve istoimene ali po pravilima C++-a preopterećene funkcije negde drugde postaje bitno koja se funkcija pri javljuje kao prva To široko otvara vrata greškama koje bi bile prektično nauhvatljive Evo primera

Pretpostavimo da smo u zaglavlju neke klase naveli funkciju **sin()** Ako se to zaglavlje učita pre poziva prave funkcije **sin** ostalak programa ce – najverovatnija – bili totalno zbunjen Zato se u verziji 2 0 (i kasnijim) uvodi tzv sigurnosno povezivanje lipa (*type-safe linkage*) Ono sakati sva imena pa redosled ucitavanja preopterećenih funkcija više nije bi tan Ta mreža dakle, hvata krupnu ribu ali kroz okca se i dalje migolji manje i dalje je moguća konfuzija ako se kroz zaglavlja redefiniše preopterecana funkcije

Listing a34

```
overload preopFunk( // priprema
int preopFunk(int)r // prva deklaracija
double preopFunk(int, int)r // druga
void preopFunk()r // treca
main(){
        preopFunk(1)r
        preopFunk(2.3)r
        preopFunk()
}
```

Preopterecivanje nepridruženih funkcija pomoću *overload* (verzija 1.2 C++ a)

DINAMIČKI OBJEKTI

U najpopularnije programske jezike npr fortran paskal i C implicitno je ugradena pretpostavka da programer unapred zna koliko će promen ljivih biti u programu kojeg će lipa one biti kada ce se stvoriti i kada će nestati Za veliku i važnu klasu programa – interaktivne programe – to ne važi Konsnik u svakom trenutku može zelražiti otvaranje novih, vecih objekata (crtaza dokumanala fontova) i program mora biti u stanju da lim zahlevima odgovori

Znatno je lakse napisati takav program ako je na nivou samog programskog jezika ugrađena neka podrska C i paskal imaju neke ne narocito elegantne mogućnosti za dinamičko stvaranje promenljivih klasični fortran ih nema Lisp i smoltok odlaze u drugu krajnost sve promenljive se stvaraju dinamički i nema posebnih oblasti programa u kojima bi sam jezik automatski brisao promenljive iz memorije U takvim jezicima promenljive zauzimaju vremenom sve vise prostora da bi u jednom trenutku ponestelo memorije Ta dva jezika obično imaju ugrađeno automatsko brisanje starih tili nepotrebnih objekata (*garbage collection*) To je za programera vrlo ugodno – programira kao da je memorija beskonačna Takva automatika nije pozeljna u svim slučajevima jer sistem presiaje da radi dok brisanje Iraje na primer automatsko prikupljanje podataka senzorima ne sme slati samo zato što je ponestalo memorije!

C++ je po „svojim" pogledima na zauzeće memorija u zlatnoj sredini U njemu je moguće dinamičko stvaranje objekata ali se ti objekti razlikuju po „duzini zivota" objekti na steku postoje samo u sintaksno određenoj oblasti (npr između zagrada { i }) dok – sa druge strane – programer može stvarati i objekte sa proizvoljnim vekom Irajanja Deo memorije ze ove druge zove se *heap* ili – u slobodnom prevodu – dinamička memorija

C++ se razlikuje od (standerdnog) paskala i C-a zato što se u njemu dinamička **promenljiva** ne samo stvara nago se obavezno i „puni" početnom vrednošću

Posledica svega ovog je da programer u C++ u mora znati više o tome šle se dogada u memoriji nego u drugim jezicima. Negativna strana tog pristupa je da stalno mora da bira između novih objekata na sleku i novih objekala u dinamičkoj memoriji Pozitivno je to što može sam de optimizuje program pa je pisanje programa koji rade u realnom vremenu sasvim mogućno u C++ u

STEK

Šla se dogada ked se izvrši ovakav deo programa

```
void prog(){
int obj = 100;
// =
}
```

Prevodilac zeuzima odgoverejući prostor na steku (tj u posebno odvojenom delu memorije koji se naziva stek) i tu postavlja broj 100 Kada kasnije naiđe na **obj** u programu prevodilac generiše masinske naredbe koje upravo sa dalinisane adresa na sleku uzimaju 100 Sve se to dešave svaki put kede se pozove funkcije **prog** nezavisno od toga gde je stek stvarno u memoriji Funkcija koje imaju samo lokalne promenljive ne zavise od ostalka programa pa ih možemo pozivati u svakom Iranutku čak i kada se one same izvršavaju U C u i C++ postoji rekurzija tj funkcija može zvati sama sebe kao u sledecem primeru

Listing a35

```
#include <stdio.h>
const n = 10;r
void rekurs(int i) {
     if (i != n)
          rekurs(++i)r
     else
          printf("kraj posla  %d!\n", i)r
}

main(){
     rekurs(0)r
}
```

Rekurzivno pozivanje funkcija

Kada se završi oblasi važenje imena promenljive tj po nailasku na odgovarajući simbol } prevodilac generise masinske naredbe koja vrh sleka poslavljaju ispod stvorenih lokalnih promenljivih čime ih istovre mano brise i oslobađa memoriju za daljnji rad U C++ u destruktor (ako postoji) bive aktiviran pre pomeranja vrha sleka

DINAMIČKA MEMORIJA

Možemo zamisliti da se stek spusla od vrha memorije nadole i da se prilikom poziva funkcije vrh sleka pomera nadole da bi se odvojio prostor za lokalne promenljive (Vrh sleka se zapravo stalno pomera nadole jer se tu beleži i povratna adresa funkcije) Ne drugom kraju raspoložive memorije na „dnu" je oblasi dinamičke memorije Odatle se uzima prostor za promenljive koje programer stvara dinamički tokom izvršenja programa U ANSI C-u bibliotečke funkcije **malloc() calloc()** i **realloc()** zauzimaju deo dinamičke memorije a funkcija **free()** je vraća U paskalu za zauzimanja ja odgovorne naredba **new()** a za vracanje

služi **dispose()**. C++ ide korak dalje: rednje sa dinamickom memorijom ulaze u samu srž ovog jezika. Službene reči **new** i **delete** ne samo što zauzimaju i oslobađaju prostor u dinamičkoj memoriji nego **new** automatski zova konstruktor a **delete** automatski zove destruktor. Zato je zvanje funkcije **new** skoro istovetno stvaranju lokalne promenljive funkcije dok je poziv funkcije **delete** skoro istovetan brisanju je promenljive kada se nađe na kraj odlasti važenja, tj. na }. Zato smo ceo proces i nazvali „dinamičko stvaranje promenljivih", umesto uobičajenijeg termina „dinamičko zauzimanje memorije".

Važno je zapaziti sklop „skoro istovetan" u gornjem objašnjenju. Razlika ipak postoji: **new** i **delete** se bave pokazivačima na objekte, a ne samim objektima. **new** vraca pokazivač na postavljeni objekat a to je, kasnije i ulazni podatak za **delete**. Ako smo svesni te razlike, možemo dinamičke objekte tretirati kao prave lokalne promenljive. Sledeći primeri ukazuju na potencijalne opasnosti.

Listing a36

```
#ifndef PRATNJA_H
#define PRATNJA_H
#include <stdio.h>
class pratnja {
    int i;
public:
    pratnja(int x = 0) {
      printf("poziv konstruktora sa argumentom %d\n", x);
      i = x;
    }
    pratnja(){
      printf("poziv destruktora za objekat br. i = %d\n", i);
    }
    void print() { printf("pratnja:: print(); i = %d\n", i);}
};
#endif PRATNJA_H
void vezba1(){
    // lokalna promenljiva tipa pratnja
    pratnja lokalnaPromenljiva(10);
    pratnja * glob = new pratnja(20);
    // pozivi pridruzene funkcije za promenljivu
    lokalnaPromenljiva.print();
    // pozivi pridruzene funkcije za pokazivac
    glob->print();
}

void(){
      printf("\nProgram A36 CPP\n");
      vezba1();
}


void vezba2(){
    // lokalna promenljiva tipa pratnja
    pratnja lokalnaPromenljiva(10);
    pratnja * glob = new pratnja(20);
    // pozivi pridruzene funkcije za pokazivac
    glob->print();
    delete glob; // delete menja kao sadrzaj podatak imati
                 // istu adresu za kojom je promenljiva
                 // i bila svojevremeno stvorena, sa new
}

main(){
      printf("\nProgram A37 CPP\n");
      vezba2();
}
```

Pracenje toka konstrukcije i desirukcije

Rezultat programa A36.CPP je

Program A36 CPP

poziv konstruktora sa argumentom 10
poziv konstruktora sa argumentom 20
pratnja:: print(), i = 10
pratnja:: print() i = 20
poziv destruktora za objekat br. i = 10

Bilo je zapaziti da je destruktor pozvan jedan jedini put i to samo za lokalnu promenljivu u trenutku kada ona nestaje jer je završenje programa došlo do }. Dakle, za promenljive na steku pozivanje destruktora je automatsko. To ne vazi za dinamicke objekte – njih programer mora snabdeti „svojeručnim" naredbama za oslobadanje memorija. U protivnom dogada ista sto i u prethodnom primeru: po zavrsetku funkcije **vezba1()** pokazivač **slob** nestaje (nalsio sa na desnu vitičastu zagradu) ali promenljiva na koju je on pokazivao – ostaje u memoriji. To je najgora moguća situacija: promenljiva ostaje zauvek u memoriji a programer više nema nacina da je ukine. Evo rešenja (podrazumeva se prisustvo klase **prainja**)

Rezultat je sada

Program A37 CPP

poziv konstruktora sa argumentom 10
poziv konstruktora sa argumentom 20
pratnja:: print(), i = 20
poziv destruktora za objekat br. i = 20
poziv destruktora za objekat br. i = 10

Problema sa **new** i **delete** ima napralak. S obzirom na uvodni karakter ovog teksta, ne možemo sa njima dalje baviti.

NASLEDIVANJE

Pračesto izgleda da je pisanje novog programa samo prepisivanja sterog, ali nekako nikad na sasvim isti način. Postojeći program može biti idealna podloga za kretanje u novi projekat, ali može biti pretežak za preradu, ili nedostaje dokumentacija. Programeri precesto „izmišljaju točak" – nikako da dođu u situaciju da iskoriste već napisane programa (svoje ili tuđe).

Na nivou programera kao pojedinca, ta nemogucnost može biti knična, a i ne mora. Na nivou jednog društva koje zapravo zavisi od softvera (a takvo je skoro svako razvijanje društvo današnjice) nemogućnost iskorišćavanja postojećeg fundusa programa košta milijarde dolara godišnje. Otkrivenje objektnog programiranja je upravo sistemski obrazlozena mogućnost da se na nivou jezika tj. uz pomoć računara koristimo vec napisanim programima. C++ to takođe dozvoljava: postoji nasledivanje klasa.

Već napisane objekte možemo svom novom programu pridružiti na dva načina:

(1) „pridruživanjem" tipova iz već postojećih klasa, i
(2) pravim nasledivanjem.

Tehnički prvi način se postiže prostim deklarisanjem interne promenljive kao pripadnika neke druge klase, dok je za nasleđivanje potrebno konstiti nove službene reči. U verzijama C++-a do verzije 1.2 postojalo je samo jednostruko nasleđivanje, tj. klesa je mogla biti naslednik samo jedne postojeće klase. U novijim verzijama 2.0 (i 2.1) klase može nasleditи nekoliko drugih klasa istovremeno – to je tzv. *višestruko nasledivanje*.

PRIDRUŽENI OBJEKTI

Pridruženi objekat je interna promenljiva klase koja je deklarisana kao objekat tj. pripadnik neke druge klase. Stvaranje takvog objekta se umnogome razlikuje od stvaranja obicnog objekta: odnosno objekta koji nije član neke druge klase. Kod običnog objekta konstruktor biva pozvan tamo gde se definiše objekat i možemo mu zadavati listu argumenata. Ako se lista ne navede, poziva se konstruktor bez argumenata (ili sa svim vrednostima već postavljenim svojedno). Kada se radi o pridruženom objektu, nikad se ne sme navesti lista argumenata u trenutku deklarisanja! Deklaracija pridruzenog objekta nalaže prevodiocu da zauzme još prostora za taj objekat u svakom primarku nove klase. Međutim, konstruktor ne može biti pozvan dok taj prostor efektivno ne buda zauzet, a to se ne dogada sve dok primerak nove klase efektivno ne bude zatrazen i dok se ne izvrši njegov sopstveni konstruktor. Konstruktori pridruženih članova moraju biti pozvani pre konstruktora nove klase.

Da bi programer to stalno imao na umu, notacija C++-a smešta te pozive *izvan* tela konstruktora nova klasa, posle liste argumenata i znaka :, a ispred otvorene zagrade. Prilikom pozivanja konstruktora pridruženog objekta koristi se iste sintakse kao i inace. Evo kako sve to izgleda u opšlem obliku:

```
class ime_klase{
//—
pridruzena_klasa ime_objekta;
//—
public:
ime_klase(argumenti), // konstruktor
//—
,
ime_klase ( ime_klase(argumenti) , ime_objekta(argumenti2)
// telo konstruktora
```

gde je **argumenti2** lista argumenata za konstruktor pridruzenog objekta: ona može ali i ne mora sadrzati argumente iz spiska **argumenti**.

Nema nikakvih ograničenja za broj pridruženih objekata u klasi. Za svaki objekat koji dodamo treba dopisati i poziv konstruktora razdvajajući ga pri tom od ostalih elemenala pomocu zareza. Konstruktori se izvršavaju redosledom navođenja u tekstu.

PRIJAVLJIVANJE GREŠAKA

Prijavljivanje grešaka je klasičan primer upotrebe spoljnje klase. Prilikom testiranja zgodno je eksplicitno prijavljivati gde se greška dogodila. Definisaćemo klasu **prijavaGreske** – ona će prijavljivati gresku – a onda cemo je uvoziti u sve ostale klase i upućivati poruke njoj umesto da koristimo uobicajene funkcije popul **printf**. Evo zaglavlja za klasu **prijavaGreske**:

Listing a38

```
#ifndef A38_H
#define A38_H
class prijavaGreske {
      char * poruka2;
public:
      prijavaGreske(char * porukaSaKlase) {
        poruka2 = porukaSaKlase;
      }
      void poruka(char * oblik, ...);
      void zavrsetak(char * oblik, ...);
};
#endif // A38_H
```

Uvoz objekata u klase

Dve navedene pridružene funkcije, **poruka** i **zavrsetak**, su primer kako se mogu pisati funkcije po svemu slične sa **printf** samog ANSI C-a. Unapred se ne zna koliko argumenata one treba da imaju, pa se u sledećem programu pozivaju posebne, ali prenosiva makronaredbe **va_list**, **va_start** i **va_end** za to. Evo programa A39.CPP

Listing a39

```
#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h> // exit()
#include "a38.cpp"
#define TEST

void prijavaGreske :: poruka(char * oblikTeksta, ...) {
      va_list argPok;
      va_start(argPok, oblikTeksta);
      fprintf(stderr, "%s ", poruka2);
      vfprintf(stderr, oblikTeksta, argPok);
      va_end(argPok);
}

void prijavaGreske::zavrsetak(char * oblikTeksta, ...) {
      va_list argPok;
      va_start(argPok, oblikTeksta);
      fprintf(stderr, "%s ", poruka2);
      vfprintf(stderr, oblikTeksta, argPok);
      va_end(argPok);
      exit(1);
}
#ifdef TEST
class stest {
      prijavaGreske greska;
public:
      stest() : greska("etest greske:") {}
      void proba(){
            greska.zavrsetak("u funkciji %s, vrednost %d\n", "proba", 1);
      }
};
main() {
      stest S;
      printf("\nProgram A39 CPP - objekat u objektu\n");
      S.proba();
}
#endif // TEST
```

Rezultat programa A39 su sledeća dve reda

Program A39 CPP - objekat u objektu
etest greske: u funkciji proba, vrednost 1

argPok je poseban pokazivač koji „prolazi" kroz celu listu argumenata. Postavljamo ga makronaredbom **va_start()** **ergPok** je ulazni podatak za funkciju **vfprintf()** a kada se ona završi, pozivom makronaredbe **va_end()** ponovo se postavlja **argPok**.

Klasa **stest** sadrži pridruženi objekat **prijavaGreske** u odliku interne promenljive **greska**. Konstruktor klase **etest** zove konstruktor za tu promenljivu. Pridružena funkcija **proba** upucuje poruku **zavrsetak** i time demonstrira jednu od velikog broja mogucnosti za prijavljivanje grešaka koja već tako prosta klasa kao što je **prijavaGreska** nudi.

Pridruženi objekti su česta pojava. Mogli bismo reći da se osobina jednog objekta pridruživanjem „pozajmljuju" jedan drugome.

PRIDRUŽENI POKAZIVAČ NA OBJEKAT

Pridruživanjem objekte već u vreme prevođenja odlučujemo se o sastavu nove klase. Ta odluka se može odložiti navođenjem pokazivača na objekat umesto samog objekta. Pretpostavimo da je definisana osnovna klasa **osnova**

class osnova { /* ... */ }

a da su još dve klase izvedene iz nje

class nasledje1 : **public osnova** { /* ... */ }
class nasledje2 : **public osnova** { /* ... */ }

Konačno, pretpostavimo da postoji klasa u kojoj se navodi pokazivač na klasu **osnova**

class sastav {
osnova * utaknut
//...
public
sastav (osnova * utak) {
utaknut = utak;
//...
}
//...
};

Pozivom konstruktora određujemo koji će objekat biti u klasi **sastav** u vreme izvršenja programa

nasledje1 X;
nasledje2 Y;
saatav c1(&X), c2(&Y)

Promenljiva **c1** sadrži objekat tipa **nesledje1**, a **c2** sadrži objekat tipa **nasledje2**

KLASA KURSOR

Pretpostavimo da smo nekako nabavili klasu za pomeranje kursora po ekranu. Evo zaglavlja te klase

Listing a40

```
#ifndef EKRAN1_H
#define EKRAN1_H
class kursor{
public:  // jednostavni pokreći kursora
         void gore(int red = 1);
         void dole(int red = 1);
         void desno(int kolo = 1);
         void levo(int kolo = 1);
              void gromestanje(int red, int stubac);  // apsolutno
pozicioniranje
         void gore_levo(); // pramentanje(1,1); }
         // brisenje delova ekrana
         void brise_ekran(); // isto sto i gore_levo
         void brise_eol(); // brise do kraja linija
         void snimanje();
         void povratak();
};
#endif // EKRAN_H
```

Klasa za kursor

U tom zaglavlju su samo „ideje", tj. koje naredbe će tu postojati. Pošto C potiče iz UNIX-a i pošto je to prilično standardizovan operativni sistem na nivou hardvera, realizacija pridruženih funkcija se zasniva na ANSI sekvencama za izmenu položaja kursora na ekranu. Evo kako bi to moglo da izgleda na PC računaru, pod pretpostavkom da se u CONFIG.SYS datoteci nalazi naredba DEVICE=ANSI.SYS. Iskoristićemo pretprocesor C-a i početni (uvak isti) niz ANSI sekvenci zapamtićemo kao ATN. Evo datoteke A41.CPP

Listing a41

```
#include <stdio.h>
#include "a40.cpp"
#define ATN "\x1b["

void kursor::gore(int redovi){
  printf(ATN "%dA" redovi);
}
void kursor::dole(int redovi){
  printf(ATN "%dB", redovi);
}
void kursor::desno(int stupci){
  printf(ATN "%dC", stupci);
}
void kursor::levo(int stupci){
  printf(ATN "%dD", stupci);
}
void kursor::gromestnje(int red, int stubac){
  printf(ATN "%d;%df", red, stubac);
}
void kursor::brisu_ekran(){
  printf(ATN "2J");
}
void kursor::brise_eol(){
  printf(ATN "K");
}
void kursor::snimanje(){
  printf(ATN "s");
}
void kursor::povratak(){
  printf(ATN "u");
}
```

Metodi za kursor

Konačno, ato i program koji to sve testira.

Program radi sasvim očekivano – pomera kursor i briše. Pretpostavimo sada da želimo ovu opštu klasu (a ona jeste opšta jer bi sve to radilo i na stotinama drugih računara i terminala, a ne samo na PC-u) da ograničimo na što bolje iskorišćavanje resursa konkretnog računara

Listing a42

```
#include "a41.cpp"
void cekanje() { for (long int i = 0; i < 1111111; i ++) {}; }
main () {
      kursor kursorc;
      cursor.brise_ekran();
      cekanje();
      cursor.promestanje(12,40); // centar ekrana
      cekanje();
      cursor.desno(20);
      cekanje();
      cursor.levo(40);
      cekanje();
      cursor.gore(4);
      cekanje();
      cursor.dole(8);
      for (int i = 0; i < 50; i ++)
        printf("X");
      cursor.levo(40);
      cekanje();
      cursor.brise_pol();
      cekanje();
}
```

Test program za kursorsku klasu (primena klase u klasi)

I konkretnog jezika. U *Turbo C++*-u postoji funkcije **gotoxy** i evo kako bismo se njom mogli da proizvedemo isti efekat

Listing a43

```
#include <conio.h>
#include "a00.cpp"
class turbo_cursor : public kursor {
public:
      void turbo_kursor :: gore(int redovi = 1)
        { gotoxy(wherex(), wherey()-redovi); }
      void turbo_kursor :: dole(int redovi = 1)
        { gotoxy(wherex(), wherey()+redovi); }
      void turbo_kursor :: desno(int stepeni = 1)
        { gotoxy(wherex()+stepeni, wherey()); }
      void turbo_kursor :: levo(int stepeni = 1)
        { gotoxy(wherex()-stepeni, wherey()); }
      void turbo_kursor :: promestanje(int row, int col)
        { gotoxy(col, row); }
      void turbo_kursor :: gore_levo () { promestanje(1,1); }
      void turbo_kursor :: brisi_ekran() { clrscr(); }
      void turbo_kursor :: brise_pol() { clreol(); }
};
```

Primer naslеdivanja klase kursor

Klasa **turbo_kursor** nasleđuje klasu **kursor** što se u prvom redu klase vidi iz prisustva dvotačke službene reči **public** i imena klase koja se nasleđuje. Glavni program A44 CPP potpuno je isti kao i prethodni glavni program, uz dva izuzetka

(1) umesto A41 sada se učitava datoteka A43 CPP jer je tu definicija klase **turbo_kursor** i

(2) umesto tipa **kursor** promenljiva **curaor** sada je tipa **turbo_kursor**

Listing a44

```
#include <stdio.h>
#include "a43.cpp"
void cekanje() { for (long int i = 0; i < 1111111; i ++) {}; }
main () {
      turbo_kursor cursor;
      cursor.brise_ekran();
      cekanje();
      cursor.promestanje(12,40); // centar ekrana
      cekanje();
      cursor.desno(20);
      cekanje();
      cursor.levo(40);
      cekanje();
      cursor.gore(4);
      cekanje();
      cursor.dole(8);
      for (int i = 0; i < 50; i ++)
        printf("X");
      cursor.levo(40);
      cekanje();
      cursor.brise_pol();
      cekanje();
}
```

Skoro potpuno isti program, ali druga klasa

Spolja gledano, program radi isto što i pre, ali razlika u načinu na koji se to postiže je ogromna. Klasa **kursor** je prenosiva ali spora, dok je **turbo_kursor** brža, ali samo za *Turbo C++* i PC računare.

Primetimo da je u glavnom programu izmena bila minimalna, a da nam za novu verziju nije bila potrebna klasa **kursor** već samo njen tekst (i odgovarajuća OBJ datoteka na disku). To otvara novu mogućnost za autora softvera isporučivanje programa koje korisnik može sam da dopiše. Podrazumeva se da je to pogodnost i za programera koji prenosi svoj program na različite softverske i hardverske platforme. Nasleđivanje klasa omogućava da se uz pomoć C++-a izmene lokalizuju i bezbolno zamene klasama za „domaći teren"

SINTAKSA NASLEĐIVANJA

Nasleđivanje se označava na sledeći opšti način

class imeIzvedeneKlase : imeOsnovneKlase {
// telo deklaracije klasa
};

Članovi klasa su tipa **private** osim ako ih eksplicitno ne označimo kao **public**. Isto važi i u kod nasleđivanja: osnovna klasa je tipa **private** ako se na navede da je tipa **public**. Iz gornje deklaracije sledi samo da svi **public** članovi osnovne klase mogu biti samo pridruženi članovi izvedene klase, ali da do njih konsnici izvedena klase nemaju pristup. Ako želimo da **public** članovi osnovne klase budu dostupni konsnicima izvedene klase, moramo iza dvotačke navesti službenu reč **public**. Zato se najčešće upotrebljava sledeći oblik nasleđivanja

class imeIzvedeneKlase : public imeOsnovneKlase {
// telo deklaracije klase
};

Izvedena klasa automatski poprima sva interna promenljiva i sva pridružene funkcije osnovne klase. Moguće je dodavati nove promenljive i nove pridružena funkcije. Unutar izvedena klase pridružena funkcije smeju zvati **public** pridružene funkcije i koristiti **public** promenljive.

Konstruktori, destruktori i preopterećeni **operator=()** se na nasleđuju automatski. Destruktori bivaju pozivani automatski po prestanku važenja imena objekta u nekom delu programa. Za konstruktore ja postupak komplikovaniji, jer oni su uglavnom preopterećeni i primaju po više argumenata. Sintaksa poziva konstruktore je zato proširena tako da se može tačno navesti koji se konstruktor iz osnovna klase poziva pre nego se izvrši konstruktor izvedene klase. Za izvedeni konstruktor opšti oblik je

imeIzvedeneKlase::imeIzvedeneKlase(argumenti1) : (argumenti2) {
// telo konstruktora izvedene klase
};

gde su **argumenti1** – spisak argumenata za konstruktor izvedene klase a **argumenti2** – spisak argumenata za konstruktor osnovne klase. Primetimo da se ne navodi ime objekta kojem je poziv konstruktora osnovne klase upućen – mada **this** uvek „pali"

Ako je konstruktor osnovne klase preopterećen, bira se onaj konstruktor koji odgovara po listi argumenata. Ako se na navede poziv konstruktora osnovne klase, podrazumeva se poziv konstruktora bez argumenata

Ako izvedena klasa sadrži pridružene objekte, njihovi konstruktori bivaju pozvani u redosledu navođenja u tekstu programa, ali uvek posle poziva konstruktora osnovne klase

U sledećem primeru vidi se lačan redosled pozivanja konstruktora za tri klase

Rezultat programa A45 CPP pod Turbo C++-om 1.0 je sledeći niz redova

Program A45.CPP

izvedenaKlasa X

osnovnaKlasa(int) poziv konstruktora
clan1(int) poziv konatruktora
clan2(int) poziv konstruktora
konstruktor izvedenaKlase
destruktor izvedeneKlase
clan2 poziv destruktora
clan1 poziv destruktora
osnovnaKlasa poziv destruktora

izvedenaKlasa2 X

osnovnaKlasa() poziv konstruktora
clan1() poziv konstruktora
clan2() poziv konatruktora
konstruktor izvedenaKlase2
destruktor izvedeneKlase2
clan2 poziv destruktora
clan1 poziv destruktora
osnovnaKlasa poziv destruktora

izvedeneKlasa3 X

osnovnaKlasa(int) poziv konstruktora
clan1(int) poziv konstruktora
clan2(int) poziv konstruktora
konatruktor izvedeneKlase3
destruktor izvedenaKlase3

clan2 poziv destruktora
clan1 poziv destruktora
osnovnaKlasa poziv destruktora

Listing a45

```
#include <stdio.h>
// makrosredstva koja generiše klase
#define generisanjeKlasa(IMEKLASE) class IMEKLASE { \
  int i;\
public: \
  IMEKLASE() | puts(#IMEKLASE "() poziv konstruktora"); } \
  IMEKLASE(int) | puts(#IMEKLASE "(int) poziv konstruktora"); } \
  ~IMEKLASE(); | puts(#IMEKLASE " poziv destruktora"); } \
}

generisanjeKlasa(osnovnaKlasa);
generisanjeKlasa(clan1);
generisanjeKlasa(clan2);
// Klasika
class izvedenaKlasa : public osnovnaKlasa {
      clan1 c1;
      clan2 c2;
public:
      izvedenaKlasa() : (1), c1(1), c2(1); |
        puts("konstruktor izvedeneKlase ");}
      }
      ~izvedenaKlasa() { puts("destruktor izvedeneKlase "); }
};

// bez eksplicitnog poziva konstruktora
class izvedenaKlasa2 : public osnovnaKlasa |
      clan1 c1;
      clan2 c2;
public:
      izvedenaKlasa2() |
        puts("konstruktor izvedenaKlasa2 ");
      }
      ~izvedenaKlasa2() | puts("destruktor izvedeneKlase2 "); }
};

// neuredjeni pozivi konstruktora
class izvedenaKlasa3 : public osnovnaKlasa {
      clan1 c1;
      clan2 c2;
public:
      izvedenaKlasa3() : c2(1), c1(1), (1) {
        puts("konstruktor izvedenaKlasa3 ");
      }
      ~izvedenaKlasa3() { puts("destruktor izvedenaKlasa3 "); }
};

main () {
printf("\nProgram A45.CPP \n");
  {
    puts("\n\t izvedenaKlasa Xs");
    izvedenaKlasa Xs;
  }
  |

  puts("\n\t izvedenaKlasa2 Xs");
  izvedenaKlasa2 Xs;
  }
  {
    puts("\n\t izvedenaKlasa3 Xs");
    izvedenaKlasa3 Xs;
  }
}
```

Redoslied izvrsavanja konstruktora i destruktora

Da bi smo „napismeno" dobili izveštaj o izvršavanju destruktora u glavnom programu stavili smo poziv svake klase u poseban par vitica stih zagrada. Ako lista konstruktora nije navedene poziva se konstruktor bez argumenata. Pozivi destruktora (osim za prividene objekte) uvek se izvršavaju obratnim redoslednom u odnosu na konstruktore

PRISTUP ČLANOVIMA OSNOVNE KLASE

Do sada smo koristili dve službene reči – **private** i **public** – da bi ograničili pristup podacima. Mogucnost nasleđivanja primorava na uvođenje još jedna takva reči – **protected**. Naime, kako učiniti dostupnim clanova osnovne klase u izvedenoj klasi a da se pritom „privatnost" podataka ne ugrozi!? Podatke koji u osnovnoj klasi treba da ostanu nepristupačni a koji istovremeno treba da u izvedenoj klasi budu prastupačni deklarišemo kao **protected**. Evo primera kako se koristi **protected**

U klasi **derived1** prevodilac prijavljuje grešku u smislu „I nije dostupno u **derived1**". Suprotno tome, u klasi **derived2** promenljiva i je deklarisana kao **protected**, pa je dostupna svim naslednicima

VIRTUELNE FUNKCIJE

Nasleđivanjem menjamo postojeću klasu ako je one „skoro" dobra za rešavanje aktuelnog problema, a samo treba malo da je „popravimo". Službenim recim **public**, **private** i **protected** štitimo podatke i funkcije u osnovnoj i izvedenim klasama. Nasleđivanje je moćan način pisanja programa, i to u mnogim primerima, ako osnovna klasa dobro radi, a nasleđena klasa na – sve je verovatnoća da su greške lokalizovane baš samo u novoj nasleđenoj klasi!

Listing a46

```
// a46 -- Ilustracija sluzbene reci protected
#include <stdio.h>
class osnal {
      int i; // i je private
public:
      base1(int ii = 0) { i = ii; }
      void put(){} | printf("base1 = %d\n", i); }
};
class base2 |
protected:
      int i) // i je private za clanove
             // ali je public za naslednike
public:
      base2(int ii = 0) | i = ii; }
      void print(); | printf("base2 i = %d\n", i); }
};
class derived1 : public base1 |
public:
// mnogo sledeceg reda za program za nama ni preveti
      void print(); | put() |"derived1 i = %d\n", i; } ;
};
class derived2 : public base2 |
      void print(); | printf("derived2 i = %d\n", i ); }
};
```

Ilustracija službene reči protected

Nasleđivanje predstavlja da postoji sistem klasa, odnosno njihova hijerarhija u obliku stabla pojmova. Takve klasifikacije su uobičajene pojave u biologiji, hemiji i drugim naukama, i zapravo se smatra da neka oblast postaje nauka tek sa pojavom strukture klasifikacije pojmova

Tehnički, za pravu primenu nasleđivanja potrebne su tzv. virtuelne funkcije. Da bismo shvatili u čemu je problem razmotrimo sledecu klasu

Listing a47

```
#include <stdio.h>
#ifndef OPIS1_H
#define OPIS1_H
class opis {
protected: // izvedene klase imaju pristup
  char * informacija;
public:
  opis(char * info) { informacija=info; |}
  void print() { printf("%s\n", informacija); }
};
#endif // OPIS1_H

U datoteci A48-CPP su dve klase, staza i kristal, izvedene iz klase
opis:
```

Prvi deo primera za virtuelne klase (klasa koja ispisuje objasnjanja na ekran)

U datoteci A48.CPP su dva klasa **sfera** i **kristal**, izvedene iz klase **opis**

Listing a48

```
#include "a47.cpp"
#ifndef OPIS2_H
#define OPIS2_H
class sfera : public opis {
      float poluprecnik;
public:
      sfera(char * info, float pola) : (info), poluprecnik(pola) {}
      // nadelinise funkcija iz osnovne klase
      void print();|
        printf("%s  ", informacija);}
        printf("poluprecnik = %g\n", poluprecnik);|
      }
};

class kocka : public opis {
      float ivica_duzine;
public:
      kocka(char * info, float iuica) : (info), ivica_duzine(ivica) {}
      // nadelinise funkcija iz osnovne klase
      void print();|
            printf("%s  ", informacija);}
            printf("ivica duzine = %g\n", ivica_duzine);|
      }
};
#endif // OPIS2_H
```

Dve klase, *sfera* i *kristal*, izvedene iz klase *opis*

Nastavija se u sledecam broju

376 [OS MSDOS, PC] Konsnicima DOS-a 4.XX koji su se opredelili za particije veće od 33 megabajta slalno je visio mac nad glavom – pojedini „stari" programi koji rade sa PSP-ovima su mogli da zabrljaju sva podatke na disku. Lak protiv loga je instaliranje programa SHARE koji se normalno koristi za višeprogramski rad. DOS 4 je čak i „bez pitanja" učitavao SHARE ako bi ga našao u *root* direktorijumu. DOS 5.0 je sa te strane mnogo bolji – potreban kod iz SHARE.a koji se bavio particijama vecim od 33 megabajta prepisan je u sam DOS, pa SHARE treba učitavati samo onda kada se zaista radi sa mrežom *DesqView*-om ili nečim sličnim!

377 [OS MSDOS, PC] Kada smo već kod DOS.a 5.0 da pomenemo još neke zgodne osobine njegove naredbe DIR za koje mnogi ne znaju. Sa DIR /S dobijamo sadržaj svih poddirektoriia tekućeg kataloga, zatim potkataloga ih potkataloga i tako dalje. Sa DIR /L dobijamo ispis malim slovima, sa DIR /A ispisujemo i atribute a sa DIR /B samo imena datoteka bez ikakvih drugih podataka. Sa DIR /Ox najzad možemo da sortiramo ispis prema imenu datoteke (DIR /ON) ekstenziji (DIR /OE) veličini (DIR /OS) i datumu kreiranja (DIR /SD). Da bi stvar bila još lepša neke od ovih opcija možemo učiniti permanentnima tako što ćemo u AUTOEXEC.BAT napisati nešto poput SET DIRCMD =/ON/L [Preuzeto sa Sezama, autor **Zoran Milosavljević** (*zormi*)]

378 [OS MSDOS, PC] Koliko godina već radimo sa DOS-om a ipak se i dalje nađe po nešto novo i neočekivano! Otkucajte u komandnoj liniji ECHO OFF i prompt nestaje! Na ECHO ON se vraća. Isprobano na DOS-u 3.30 i DR DOS-u 5.0 verovatno važi i u svim ostalim verzijama. Baš ludo, zar ne? [Preuzeto sa Sezama, autor **Kiro Velkovski** (*kvelkovski*)]

379 [MREZ.NOVELL, PC] Ako imate potrebe da u nekoj aplikaciji koja radi pod *Novell*-om otvorite više od 40 datoteka, treba da kreirate konfiguraciju SHELL.CFG i u njoj napišete FILE HANDLES = 100 ili koliko već. Tajlova i Erba u isto vreme otvorili. [Preuzeto sa Sezama, autor **Jozef Piri** (*jpiri*)]

380 [MREZ.NOVELL, PC] Kada već kreirate datoteku SHELL.CFG, dodajte u nju i SHOW DOTS = ON kako biste na primer u *Windows* menadžerima imali i oznaku za prelazak u „roditeljski" katalog [Preuzeto sa Sezama, autor **Zoran Milosavljević** (*zormi*)]

381 [MIKR] Na slici 1 je prikazan kratak program koji treba asemblirati (potreban vam je neki asembler koji podržava 80386). Linkovali i startovali. Ako prijavi *POPAD fails test* vas 80386 ima bag u POPAD instrukciji.

Mala anketa na Sezamu pokazuje da je **nemoguće** jedino pronaći 80386DX bez baga – našlo se 80386SX sa i bez baga kao i „bagovitih" i nebagovitih" 80486. Zanimljivo bi bilo znati posloji li bag i kod 386 klonova (što bi pokazalo da li su njihovi konstruktori za sta pisali svoj mikrokod ili su „krali" od *Intela*) ali do sada nismo imali priliku da startujemo test na nekom od njih.

382 [ASEM, PC] Na slici 2 prikazan je kratak (170 bajta) asemblerski program koji omogućava potpunu kontrolu *Lock* indikatora na PC tastaturi. Program bi trebalo da radi bez obzira na eventualno posloianje drajvera za tastaturu.

[Preuzeto sa Sezama, autor **M. Mihajlović** (*mmihajlovic*)]

Nastavlja se na strani 69

Listing 2

```
;
;       Program za kontrolu Num, Caps i ScrollLock tastera
;       Autor: M.Mihajlović
;
;       assyllo:  XADM LOCK_ASM  ili  MASM LOCK_ASM;
;                 TLINK LOCK /t  ili  LINK LOCK /T;
;
;   Napomena: Program će kontrolše Shift status preko adrese 0000:0417h
;             Iako ličnо nimа imao problema sa sinhronizacijom Lock
;             indikatora (na adresi 0000:0410h) treba ga uključiti komentar
;             liniju sa int 08h čime se vrši update shift statusa
;
;
lock_offset     equ     17h                     ; Lock status offset

code            segment
                assume  cs:code, ds:code

                org     100h

main            proc    far
start:          mov     ax,ds                   | ne i da se ne luti
                mov     dx,ax
                mov     ds,40h                  ; Lock status bajt u ds
                mov     es,ax
                cmp     al,01h                  | Prvi char argv-a
                mov     dl ds:lock_offset
                mov     al 0e0h                 ; Ucitaj argv-a
                xor     dh,dh                   ; dh=0
                cld
                cmp     al,0
                je      No_argv                 ; Ako nema argumenta ili
                mov     cx cx                   ; nije (ako drugačiji)

Start_loop:                                     ; ovde počinjemo petlju za
                                                | obradu argv[]
                lodsb                           ; stavlja [si] u al [and tam argv]
                cmp     al,20h                  |
                je      End_loop                |
                                                ; Ova leb je zbog uniformnosti
if_Caps:        cmp     al,43h                  | C
                je      Caps
                cmp     al,63h                  ; c
                jne     if_Num
Caps:
                mov     dh,40h                  ; CapsLock=ON
                jmp short End_loop
if_Num:
                cmp     al,4eh                  | N
                je      Num
                cmp     al,6eh                  ; n
                jne     if_Scroll
Num:
                mov     dh,20h                  ; NumLock=ON
                jmp short End_loop
if_Scroll:
                cmp     al,53h                  | S
                je      Scroll
                cmp     al,73h                  | s
                jne     Plus
Scroll:
                mov     dh,10h                  | ScrollLock=ON
                jmp short End_loop
Plus:
                cmp     al,2bh                  | +
                jne     Minus
                or      dl,dh
                jmp short End_loop
Minus:
                cmp     al,2dh                  | -
                jne     No_argv
                mov     ah,dh
                not     ah                      ; Ako je zadat minus xLock-OFF
                and     dl ah
```

```
End_loop:
                loop    Start_loop              ; Loop ako je cx > 0
                mov     es:lock_offset,dl       ; Upisujemo nove lock statuse
                                                ; u low byte (0040:0017-dl)
                                                ; Uklonili kodantar ispred
                                                ; sledeće linije ako postoje
                                                ; problemi sa Lock indikatorima
                int     09h                     ; Refresuje tastaturu
                int     20h                     ; Kraj programa

No_argv:
                                                ; Nema argumenata ili su pogrešni
                mov     dx,offset Msg
                mov     ah,9
                int     21h                     ; Dos funkcija za display stringa
                int     20h                     ; Kraj programa

Msg             db      'Koristi MS C4 i 6+ za regulisanje Num, Caps
main            endp    i ScrollLock-a.$

code            ends
                end     start
```

Program za kontrolu *Num, Caps* i *ScrollLock* tastera

Listing 1 BUGTST.ASM

```
                386
;
; BUGTST.ASM - By: John Lauro
;              Based on bug found by Jeff Prothero
;
DATA    SEGMENT  USE16 PUBLIC
MSG     DB       'POPAD ok ',13,10, $
MSGTOK  DB       '*** POPAD fails test *** ',13,10 '$
DATA    ENDS
;
;
STACKSG SEGMENT  WORD STACK
        DW       100 DUP(?)
STACKSG ENDS
;
;
CODE    SEGMENT  USE16 PUBLIC
        ASSUME   CS:CODE, SS:STACKSG, DS:DATA
CODES   PROC     FAR
START:  MOV      AX,DATA
        MOV      DS,AX

        mov      eax,12345678
        mov      edx, 0
        mov      edi, 0
        pushad
        popad

; The instruction immediately following popad is the critical
; instruction.  Simple fix, insert a NOP after popad

        mov      ecx, [edx+edi]

        cmp      eax, 12345678
        je       ok
        mov      dx, offset MsgTok
outmsg: mov      ah  09h     | ConStrlog=output
        int      21h

        mov      dh, 4ch     | Exit
        mov      al, 0
        int      21h

ok:     mov      dx, offset msg
        jmp      outmsg

CODES   ENDP
CODE    ENDS
        END      START
```

Program za otkrivanje baga u mikroprocesoru I386

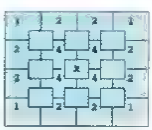
NAJMANJI OD NAJVEĆIH

Letnji meseci i relativno teška pitalica učinili su da odziv bude nešto ispod uobičajenog. Dobili smo, ipak, nekoliko veoma interesntnih programa i to, skoro bez izuzetka, posredstvom Sezama. Znači li to da se naše pitalice sve više sele u domen elektronske pošte?

Podsetimo se, pre svega, problema: svi celi brojevi između 1 i 16 su upisani u polje tabele - jedan broj u jednom kvadratiću. Zatim je u svaki kružić upisan zbir podataka sa četiri polja koja ga okružuju. Dobijenih 9 zbirova su sortirani po veličini, a onda su odbačena dva najveća. Trebalo je rasporediti početne brojeve tako da sledeći (treći) **najveći** zbir bude **što manji**.

Zadatak se, kao i mnogi drugi, može rešiti primenom tzv. „grube sile", tj. isprobavanjem svih mogućih rasporeda brojeva 1-16 što jeste popriličan ali ze današnje personalne ne i nedostižan posao. Pe ipak, ovoj su rubrici primereniji programi koji se zasnivaju na detaljnijoj analizi problema i korišćenju kompjutera tek u nekoj od kasnijih faza rešavanja.

Za početak ćemo u svako polje tabele upisati broj suma u kojima to polje figuriše. Dobijamo tabelu sa slike 1. iz koje se vidi da su u četiri središnja polja upisane četvorke. U la polja je, dakle, poželjno upisati što manje brojeve koji će, kao takvi, „učestvovati" u vecem broju zbirova pa je sloge prilično verovatno da će središnja suma (ne slici 1 označena sa X) biti najmanja.



Slika 1. U koliko zbirova svako polje učestvuje

Ukoliko je središnja suma najmanja, najveće dve sume biće negde u osam spoljnih kvadratića. Ako zanemarimo simetrične, dobijamo svega 6 različitih rasporeda tih najvećih suma. Za svaki od tih rasporeda napravimo novu tabelu (slika 2), u čija ćemo polja upisivati u koliko suma dotično polje učestvuje, ali ne uzimajući u obzir središnju kao i dve najveće sume. Slovom W je označena jedna od dve najveće ili najmanja suma.

Sada za svaki od ovih šest rasporeda izračunamo sumu svih suma (ne uzimajući u obzir dve najveće kao ni najmanju), pridržavajući se pravila: ako polje učestvuje u više zbirova u njega će biti upisan manji broj. Tako dobijamo

a) 0*(16+15) + 1*(14+13+12+11+10+9) + 2*(8+7+6+5+4+3) + 3*(2+1) = 144
b) 0*(16+15) + 1*(14+13+12+11+10+9) + 2*(8+7+6+5+4+3) + 3*(2+1) = 144
c) 0*16 + 1*(15+14+13+12+11+10+9) + 2*(8+7+6+5+4+3+2) + 3*1 = 157
d) 0*(16+15) + 1*(14+13+12+11+10+9) + 2*(8+7+6+5+4+3) + 3*(2+1) = 144
e) 1*(16+15+14+13+12+11+10+9+8) + 2*(7+6+5+4+3+2) + 3*1 = 165
f) 1*(16+15+14+13+12+11+10+9) + 2*(8+7+6+5+4+3+2+1) = 172

Ako sada te zbirove podelimo sa 6 (jer u njima učestvuje šest suma) dobijamo

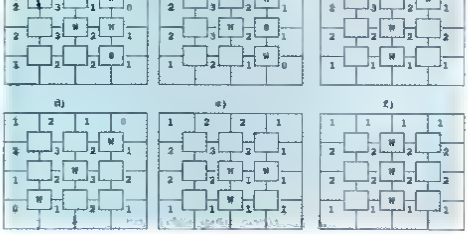
a) 24 b) 24 c) 26 167 d) 24 e) 27 5 f) 28 67

Vidimo da je u slučajevima *c*, e i *f* treća najveća suma bar 27, 28 i 29, a u preostalim primerima je bar 24. To znači da je treća najveća suma u najboljem slučaju 24 i to pod uslovom da su sve sume (osim središnje najmanje i dve odbačene najveće) jednake 24.

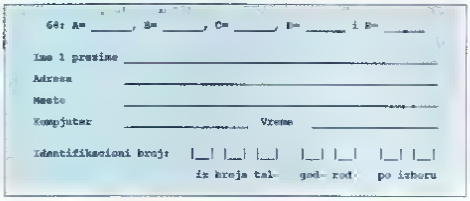
Ako sada pogledamo slučajeve *e* i *b*, vidimo da u sumi označenoj sa Ž učestvuju dva polja sa jedinicom i dva polja sa dvojkom. Pošto u polje sa manjom cifrom upisujemo veći broj, te dve sume bi u najboljem slučaju bile 10+9+4+3=26, što je više od 24, tako da ni u tim slučajevima ne možemo dobiti optimalno rešenje.

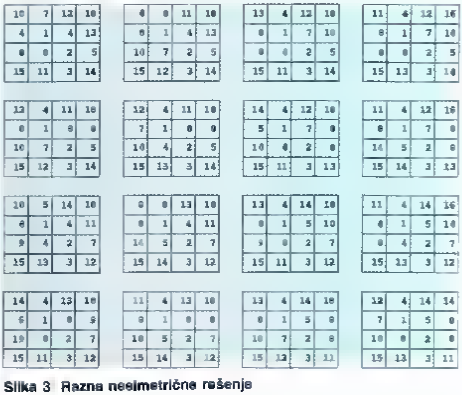
Ostaje nam još slučaj *d* kod koga, podržavajući se izloženih pravila, začas dolazimo do rešenja. Ta brzina i nije neočekivana kada se uzme u obzir da takvih rešenja (kada se zanemare simetrične) ima čak 66, što je ujedno i redni broj ove pitalice. Na slici 3 pnkazano je dvadesetak od tih 66 rešenja u nešto „kompnmovanom" obliku.

Do rešenje se može doći i primenom programa sa slike 4 koji je zajedno sa detaljnim obrazloženjem nalik na prethodno doneo prvu nagradu (2000 dinara) *Slavku Kukriki* iz Ljubljane (*skukrika* na Sezamu) – program se kao što ćete videti kada ga otkucate izvršava praktično trenutno, tj. troši vreme jedino na samo ispisivanje rešenja. Svi ostali pnmljeni programi su se izvršavali mnogo mnogo duže ali je i u njima bilo veoma zanimljivih ideja. Jedna od njih donela je drugu nagradu (1500 dinara) *Milovanu Kovačeviću* iz Sremske Kamenice (*mejki* na Sezamu) čiji program pnkazuje rešenja interaktivno, onako kako ih pronalazi. Treću nagradu (1000 dinara) smo izvukli – najviše sreće imao je *Mano Škrtić* iz Karlovca. Pohvale su zaslužili *Boško Kopnvica*, *Miloš Prvulović* (*prvul*) i *Aleksander Zimonjić* a ze godišnje lekmićenje se boduju rešenja zaključno sa 26 – iako je pravo rešenje 24 primili smo i kupone u koje su upisani mnogo veći brojevi!



Slika 2. Šest mogućih rešenja





Slika 3. Razna nesimetrične rešenja

Listing 4

```
program Pitalica_55;
{

 Prama programu Slavka Kukrika
        "Računari 75"

}
var psarray [1..6] of integer;
    s:array [1..16] of integer;
    all:integer;

function sani(pos:integer):boolean;
var i:integer;
    rezultat:boolean;
begin
  rezultat:=true;
  for i:=1 to pos-1 do if p[pos]=p[i] then rezultat:=false;
  sani:=rezultat;
end;

procedure trazi
var f,fi:integer;
    q:array [1..16] of boolean;
    sum:boolean;
begin
  a[2]:=p[1]+2; a[5]:=p[2]+2; a[7]:=p[3]+2;
  a[10]:=p[6]+2; a[12]:=p[5]+2; a[15]:=p[6]+2.
  a[1]:=24-a[2]-a[5]-a;
  a[3]:=24-a[2]-a[7]-2;
  a[8]:=24-a[7]-a[12]-1;
  a[13]:=24-a[5]-a[10]-2;
  a[14]:=24-a[10]-a[15]-1;
  a[16]:=24-a[12]-a[15]-1;
  for f:=1 to 16 do q[f]:=false;
  for f:=1 to 16 do q[a[f]]:=true;
  sum:=true;
  for f:=1 to 16 do sum:=sum and q[f];
  if sum then
  begin
    all:=all+1;
    for f:=0 to 3 do
    begin
      for fi:=1 to 4 do write(a[fi+4*f],' ');
      writeln;
    end;
    writeln;
  end;
end;

procedure permut(pozicija:integer);
begin
  p[pozicija]:=1;
  repeat
    if sani(pozicija) then
      if pozicija=6 then trazi
                    else permut(pozicija+1);
    p[pozicija]:=p[pozicija]+1;
  until p[pozicija]>6;
end;

begin
  a[4]:=16; a[13]:=15; a[6]:=2; a[11]:=1;
  all:=0;
  permut(1);
  writeln(' Postoji ukupno ', all, ' realistiih mesimetriinih resenja ');
end.
```

Slika 4. Program koji rešava problem

NAJMANJI KVADRAT

Evo još jednog zadatka na temu optimizacije. Pronađite međusobno različite prirodne brojeve A, B, C, D i E takve da je $A^2+B^2=C^2=D^2+E^2$ (zbir kvadrata brojeva A i B jednak je zbiru kvadrata brojeva D i E odnosno kvadratu broja C). Pri tome se zahtijeva da broj C bude što manji.

Rešenja pitalice, zajedno sa **obrazloženim** programima koja ste napisali, pošaljite na uobičajenu adresu: „Računari" (za Dejanova pitalica), Bulevar vojvode Mišića 17, Beograd, tako da pristignu **pre** 1. oktobra 1991. Sva pisma sa korektnim rešenjima konkurišu za novčane nagrade od 2000, 1500 i 1000 dinara, dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete **poslednja** tri cifre nečijeg broja telefona, u sledeće dva godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj. Osim običnom, rešenja ove pitalice možete da pošaljete i elektronskom poštom posredstvom Sezama – koristite menije **REDAKCIJA, PITALICE, REŠENJE** ili, iz komandnog moda, pošaljite ličnu poruku korisniku **pitalice** (*mail write pitalice*). Sva rešenja pristigla elektronskom poštom ravnopravno sudeluju u izvlačenju nagrada i konkursu za **specijalnu nagradu** od 1500 dinara, koja će biti dodeljena najboljem rešenju koje je dostavljeno modemom!

VIŠE OD EGZOTIKE

Nastavak sa strane 33

Primer na listingu 3 definiše klasu **Tecka** u modulu **Tecka0**, i podklasu **Tacka3D** u modulu **Tacka3_Tacka3D** nasleduje klasu **Tacka** tako što joj dodaje i treću koordinatu. U skladu sa tim menjaju se i nasleđeni metodi **NovaTacka** i **DuzinaTacka**.

Ovo je dovoljno za pravo objektno programiranje na nivou jezika, ali – naravno – na programeru je da počne da razmišlja u terminima objekata i da tako dizajnira program od samog početka.

PRIMER IZ ŽIVOTA

Što veći i značajniji program to je bolje pisati ga na modulu 2 a ne recimo na kliperu. Da bismo stekli kriterijum treba se postaviti u položaj krajnjeg korisnika: koliko je za njega značajno da se program izvršava brzinom munje ili da satima i danima? To pitanje nije nimalo akademsko: sticajem okolnosti u jednoj firmi je napravljen isti program (obračun plate) na *Clipper*u i na *TopSpeed Modulu 2*. Program na *Clipper*u prevlači je tako da imitira postojeći demo program na modulu ali razlika u brzini je drastične proračun plate (za 470 radnika) na *Clipper*-u traje sat i deset minuta a na modulu 2 trideset sekundi (u pitanju je AT računar običnih karakteristika). Pri tome je *Clipper* koristio svoj uobičajeni indeksni sistem pristupa podacima na disku, dok je u modulu-2 urađena totalna optimizacija upotrebe memorije (mada bez korišćenja *expanded* memorija). Sa druge strane, program na *Clipper*u je napisan za nekoliko dana, a program na modulu je pisan nedeljama.

Plate su obračunavaju mesečno i da li je zaista bitno da se program za obračun plate izvršava 100 do 150 puta brže? Pa, jeste bitno jer svaka izmena podataka u *Clipper*u znači čekanje od preko sata. Sada, zahvaljujući modulu, rukovodstvo firme može eksperimentalno da odredi tačnu vrednost boda za svaki mesec.

PRESUDA

Najavljena je i verzija 3.0, ali se poboljšanja odnose na programiranje za *Windowse* i još bolju saradnju sa ostalim jezicima iz *TopSpeed* serije. *TopSpeed Modula 2* je teška artiljerija i bez obzira na verziju predstavlja odličan sistem za pisanje programa na PC računarima.

BAJTOVI LIČNE PRIRODE

Nastavak sa strane 67

383 [OS UNIX, PC] Ako na disku držite i DOS i SCO Unix (ili Xenix) često se ukazuje potrebe za startovanjem sistema pod jednim ili drugim. Stalne promene u FDISK-u su veoma naporne, ali na sreću postoji i savršeno komforno rešenje. Treba **jednom za svegda** proglasiti *Unix* particiju aktivnom a zatim po svakom dizanju sistema na prompt *Boot* odgovoriti bilo pritiskom na Enter (u kom će slučaju biti startovan *Unix*) ili kucanjem reči DOS i pritiskom na Enter u kom slučaju sistem startuje pod DOS-om. Ukoliko želite da podignete *Unix* sa nekim posebnim *kernel*-om otkucajte njegovo ime: spisak dobijate kucajući upitnik.

Nevolja je jedino što Unix ne podržava *extended* DOS particije. [Preuzeto sa Sezama, autor **M. Aleksić** (*maleksic*)]

384 [MREZ.NOVELL.354, PC] Pomenuli smo da se pritiscima na *Ctrl-Esc* ne mogu menjati taskovi na stanicama pod programom RCONSOLE. Zaista ne mogu, ali ih funkciju obavlja plus sa numeričke tastature. [Autor: **Vanja Brčić** Poreč]

385 [IGRE, PC] Secate li se stare dobre *Elite* koja je najpre napisana za *Acorn*-ov BBC a zatim prerađena za većinu drugih kućnih računara, dakle za *Spectrum*, *Commodore*, *Amstrad*, *Ateri*? Naravno da ni PC nije izuzetak ali prva PC verzija *Elite* nije bila baš slavna pošto je korektnom brzinom radila jedino na *Amstrad PC*-ju za koji je i pisana. Odnedavno možemo da uživamo u novoj verziji *Elite Plus* koja po mnogo čemu liči na prethodnu ali nudi bolji prikaz (VGA i MCGA mod), korektnu brzinu, senčene objekte, jednostavnije komande, detaljniji je podatke o planetama (računajući i „slike" stanovnika).

Obzirom da se još od BBC-ja u ovoj igri „varalo", nema razloga da od te prakse odustanemo. Elem, učitajte svoj CDR fajl u neki HEX editor (npr. *Norton Utilities*), zatim željeni broj kredita upišete na pozicije 32-42 kao niz najobičnijih ASCII cifara, onda taj broj pomnožite sa 10 i pretvorite u heksadekadni oblik (*unsigned long integer*) pa upišite na pozicije 140-143 i to u **obrnutom** redosledu. Snimite datoteku na disk, startujte *Elite* i... bogati ste! [Preuzeto sa Sezama, autor **Zoran Milosavljević** (*zorm*)]

Pošaljite za ovu rubriku šaljite na adresu „Računari" (za „Bajtova lična priroda"), Bulevar vojvode Mišića 17, Beograd ili preko Sezama u vidu lične poruke uredniku (*mail write dejan*).

C SAMO DOBAR VIC?

U izjavi kojom su zapanjili kompjutersku industriju Ken Thompson, Dennis Ritchie i Brian Kernighan su priznali da su Unix i C, koji su oni kreirali, samo provoaprilska šala koje živi već 20 godina. Govoreći na simpozijumu posvećenom razvoju software pod Unix-om, Tompson je izjavio

"Radeći na razvoju prve verzije paskala profesora Nichlaus Wirta u ETH laboratoriji u Švajcarskoj bili smo impresionirni njegovom eleganinom jednostavnošću i snagom. Denis je upravo završio čitanje parodije na Tolkinovog „Gospodara prstenova" Kao šalu odlučili smo da napravimo parodiju ne Mullix okruženja i paskal. Denis i ja smo preuzeli operativni sistem. Gladajući Mullix dizajnirali smo novi sistem koji je bio maksimalno kriptičan i predviđen da frustrira korisnike. Nazvali smo ga Unix kao parodiju na Mullix. Denis i Brajen su radili ne stvarno deformisanoj verziji paskala, koju smo nazvali „A" Kada smo primelili da osteli zaista pokušavaju da pišu stvarne programe ne ovom jeziku brzo smo dodali još kriptičnih osobina i razvili B potom BCPL i konačno C Zaustavili smo se u trenutku kada smo uspeli da kompajler bez greške preveće sledeću sintaksu

```
for(;P(" \n"),R=;P(*|*");)for(;e=C;e=;
P("_"+(*u++/8)%2))P(*| "+(*u/4)%2),
```

Nija nam ni na kraj pamati palo da bi programeri mogli prihvatiti jezik koji dozvoljava ovakve naredbe! U svakom slučaju, Brajan, Denis i ja radimo isključivo na paskalu i mekintoš računarima. Poslednjih nekoliko godina zaista osećamo grizu savesti zbog haosa, konfuzije i stvarno lošeg programiranja koje smo proizveli našom šalom pre mnogo godina."

Vodeći proizvodači Unix i C-a odbijaju za sada da komentarišu ovu izjavu. Bodarid, vodeći proizvođač popularnih paskal i C alata, izjavio je da je od samog početka sumnjao da je ovako nešto moguće. Ubuduće će se polpuno posvetiti razvoju svojih paskal proizvoda i napustiti dalje napore oko razvoja C-a. IBM-ov predstavnik je izjavu prokomentarisao nekontrolisanim napadom smaha, ali je potom najavio odlaganje konferencije za štampu posvećenu budućnosti RS 6000 mašine najavljujući da će „VM operativni sistem biti uskoro dostupan na ovim mašinama"

Ovo je dalimičan prevod teksta koji se pojavio na Sezamu prenosom sa strane mreže. Ne nemojte da brišala diskete sa C kompajlerom i da skidate prašinu sa paskal disketa – tekst je objavljen 1 aprila. To što vas je njegov sadržaj, ako ste ljubitalj C-a, na trenutak zabrinuo (a mnoga na Sezamu jeste!) samo govori o tome da u svakoj šali ima i malo zbilje.

STRING FUKKCIJE

Već je postao običaj da se u ovoj seriji bavimo ovom ili onom string funkcijom. Svojevremeno smo se okomili na **strncpy** što je trebalo da nam posluži kao dobro iskustvo – napoznavanje nekog detalja funkcionisanja ovih bibliotečkih funkcija kod C-a, koji ni na koji način na proverava prekoračenja granice niza ili vanjanost adresa, ima uvek neprijatne posledice. Ipak, dejstvo ovih funkcija izgleda lako jednostavno i očigledno da sam opel nalaleo na sličan problem.

U jednoj petlji programa u kojoj se analizira sadržaj stringa, pored niza uslova za izlazak pojavljuje se i provera da li znak na kome se trenutno nalazimo spada u skup znakova koji je definisan drugim stringom. Prvi put kada naidamo na znak koga u skupu nema, petlja se napušta. Za ovo idealno može poslužiti funkcija **strchr** sa sledećom definicijom

char *strchr(char *string, cher c),

Funkcija vraća pointer ne prvu pojavu znaka **c** u nizu **string**, ili NULL pointer ukoliko takvog znaka nema.

Program se, naravno, ponašao polpuno „neurotično" i lak uz pomoć debugera sam otkrio jednu zanimljivu osobinu **strchr** funcija koje je, naravno, dokumetnovana – i nul terminator ulazi u skup znakova koji se proverava. Tako se desilo da je i nula na kraju stringe uvek spadale u skup zadatih znakova, pa je petlja nastavljala ispitivanje i van granice stringa. Dakle, sa **str** funkcijama iznenađenjima nikad kraja.

Kada smo već kod stringova, evo jedne česte situacije u programima koja se može rešiti samo jednom **str** funkcijom. Varujem da ste više puta sreli situaciju kada podatak koji korisnik daje treba uporediti sa labelom ispravnih vredosti. Primera ima bezbroj, od proste provere da li je korisnik odgovorio sa D. Da, N ili **Ne** (ukoliko se ovo naravno, ne rešava pritiskom na jedan laster) do interepiara nekog komandnog jezika u koma se proverava ispravnost unete komande. Generalno problem sa svodi na proveru da li je neki string jednak bilo kome iz niza unapred definisanih stringova. Ocigledno, rešenje je

```
char *moguce[] = { "rec1", "rec2", "rec3", ... , NULL};
char *check(char *translen)
{
    int i = 0;
    while (moguce[i]!=NULL && strcmp(moguce[i],translen))
        ++i;
    return( moguce[i] );
}
main()
{
    char word[128];
    gets(word);
    if ( check(word)==NULL )
    { /* ne postoji */
    } else
    { /* postoji */ }
}
```

Nišla neobično puno puta viđeno. U situacijama kada niz **moguce** treba da bude dinamički alociran (skup naredbi koji su definisane u konfiguracionoj datoteci, na primer) kod dodalno raste – treba se ili odlučiti za maksimalne dimenzije niza (oba i broj dozvoljenih reči i duzinu najduže) ili izabrali dinamičku alokaciju. U oba slučaja, učitavanje niza, alokacija prostora itd je posao koga ćete se bez razmišljanja prihvatili ako razvijate komandni interepter jer će to biti semo mali deo koda koji se rutinski može ispisati.

Pogledate sada sledeću situaciju. U realizaciji bilo kakvog komunikacionog programe koji koristi modem potrebno je proveriti kakav je odgovor modema u nekim situacijama. Odgovor modema mogu biti u obliku unapred definisanih stringova kao što su, recimo „BUSY", „NO CARRIER", „CONNECT" itd. Pošto nisu svi modemi istl. facne vrednosti ovih stringova se obično definišu u posebnoj konfuguracionoj datoteci. Iz prethodnog primera možete zaključiti koliki je posao da se jedan ovakav niz modem odgovora koji je zapisan u datoteci učita u malo monjski niz i potom po njemu pretrazivanjem pronađe da li poslednji odgovor modema jeste među njima. 20-30-40 linija koda tek da bismo zaključili da li je odgovor modema jedan od recimo pomenula tri.

U ovakvim situacijama sledeće rešenje je dalako eleganinije

Ideja je da sa reč koju treba pronaći u nizu uokviri nekim znakom (koji se inače ne može naći u samim stringovima) i da se potom jed-

```
char *moguce = "|rec1|rec2|rec3|";
main()
{
    char word[128];
    int i;
    gets(word+1);
    i = strlen(word+1);
    word[0] = word[i] = '|';
    word[i+1] = 0;
    if ( strstr(moguce,word)==NULL )
    { /* ne postoji */
    } else
    { /* postoji */ }
}
```

nom string funkcijom (**strstr**) proveri da li takav niz znakova postoji u ranije zadatom nizu. Pomenula funkcija traži prvu pojavu stringa **word** u nizu **moguce** i vraća ili pointer ne nju, ili NULL pointer ukoliko ga ne ponađe. Posebna pogodnost ovakvog rešenja je što se niz **moguće** u ovoj formi može izuzetno lako „jednim pofezom" učitati iz konfiguracione datoteke. Mana rešenja je što je ovakvo pretrazivanja sporije, ali niz **moguce** treba zaista da bude velik da sa ovaj nedostatak pokaže kao prepreka.

KOJI JE SUTRA DAN?

Odgovor na pitanje „koji je danes datum i koliko je sati" se u programima za PC resava na različite načine. Kada je datum u pitanju, koristi se odgovarajuca DOS funkcija, dok se za vreme često koriste BIOS pozivi. Kada se pak krene u neku aritmetiku sa vremenom popul „koji je datum 30 dana od danas" obično posežemo za nekim od proverenih funkcija iz nase mladosti, jer je ovaj problem bio omiljen kod programera još od kako sa računanje bionima pokazalo kao izuzetno sredstvo za „izazivanja dužnog poštovanja prema onom što radite od strane skeptičnih laika"

C kao posledicu prirode okruzenja u kome je nastao (unix operativni sistem) poseduje niz funkcija posvecenih ovom problemu. Njima se duduše, ne moze izracunati bionitem jer se uglavnom bave sadašnjošcu i budućnošcu (merenje vremena počinja od 1.1.1970) ali je malo verovatno da ce vam nešto više zatrabati za najveći broj aplikacija.

Odmah da počnemo jednom osobenošcu ovih funkcija – sistem je dizajniran lako da vodi računa o vremenskim zonama. Ako u nekom trenutku zatrazite koliko je sati i polom dobijete vrenosi (koja ima za sada ostanimo na tome binarni format) prosledite nekome u Americi i on od njc formira ascii formu dobiće svoje lokalno vreme. Naravno da bi ovo bilo moguce, potrebno je da na oba računara bude definisana vremenska zona. Iako da sistem može da izvrsi neophodne korekcije. Resenje se zasniva na tome da je binarna vrednost trenutnog vremena uvek prema GMT u (Greenwich Mean Time), a na svakom računaru se konigu ja prama vremenskoj zoni u kojoj se nalazi.

Osnovna funkcija je **time**

time_t time(time_t *timeptr),

Ona vraca broj do log trenutka proteklih sekundi od 1.1.1970 u 00:00 i to za GMT vremensku zonu (**time_t** je tipa **long**). Funkcija **localtime** će na osnovu vrednosti koju vrati **time** popuniti strukturu **struct tm** se sledecim poljima

tm_sec – sekundi
tm_min – minuti
tm_hour – sati (0-24)
tm_mday – dan u mesecu (1-31)
tm_mon – mesec (0-11)
tm_year – godina – 1900
tm_wday – dan u nedelji (0-6 0 = nedelja)
tm_yday – dan u godini (0 – 365)
tm_isdst != 0 – ako važi letnje vreme, inače 0

Funkcija **gmtime** će uraditi isto, ali računajući da lražimo GMT vreme. Funkcija **asctime** će ako joj se prosledi ova struktura, dati vreme u ascii formatu. Što će isto učiniti i **ctime** koja kao parametar koristi direktno vrednost dobijenu funkcijom **time**. Obe koriste američki format, na primer

Wed Feb 04 03:30:00 1991

što u našim uslovima nije mnogo korisno, ali se lako mogu napisati odgovarajući domaći ekvivalenti. Konačno, funkcija **mktime** će uraditi inverzan proces – ako joj prosledite popunjenu **struct tm**, izračunaće **time_t** vrednost. Strukturu **tm** ne morate popuniti podacima u normalnom opsegu – vrenosti će biti ispravno izrebrane i nakon proračuna svedena na prave vrednosti.

```
#include <time.h>
#include <stdlib.h>
#define max  64000L /* sec/dan */
time_t sada;
time_t programi;
struct tm *t;
char input[128];
char *dani[] = { "Pon", "Uto", "Sre", "Cet", "Pet", "Sub", "Ned" };
main()
{
    time(&sada);
    printf("Kod dana: ");
    gets(input);
    programi = sada + atol(input)*max;
    t = localtime( &programi );
    printf("Program se liti aktivira %d.%d.%d (%s) u %02d:%02d\n",
            t->tm_mday, t->tm_mon+1, t->tm_year+1900, dani[ t->tm_wday ]
            t->tm_hour, t->tm_min);
}
```

Objašnjenje kojim smo počeli predstavljanje **time** funkcija, u uslovima MS DOS operativnog sistema koji o vremenskim zonama ne vodi računa donekle komplikuje atvar. Jedini oslonac računanja vremena na PC-u je časovnik koji vodi BIOS i koji naravno pokazuje lokalno vreme. Zato funkcija **time** više neophodnu korekciju da od tog lokalnog vremena dode do GMT. Korekcija se viši prema unapred zadatoj konstantni koja se (kod MSC-a) odnosi na pacifičko vreme (PST: 8 sati razlika u odnosu na GMT). **Time** zato uvek ima korigovanu vrednost, a **localtime** viši konverziju unazad čime opet slize od onoga što pokazuje naš BIOS sat. Jako da na prvi pogled deluje da ova konverzija nema nekog efekta i da ja možemo zanemariti. Na žalost nije tako zbog promene letnje/zimsko vreme koja se lakoće inicijalno uzima u obzir, ali opet prema pacifikom standardu. Zato se kod korišćenja funkcije **mktime** može pojaviti razlika od jednog sata. Problem se može rešiti **environment** promenjivom **TZ** i korišćenjem **tzset()**. Funkcija **TZ** vanjabla treba da ima formal **TMZnDST**, gde je **TMZ** oznaka vremenske zone, **n** razlika u satima u odnosu na GMT i **DST** (opciono) zona čija se promena letnje/zimsko vreme koristi. Funkcija **tzset()** koji treba izvesti negde na početku programa konsultuje ovu promenjivu i na osnovu nje podešava nekoliko statičkih varijabli koje se koriste pri preračunavanju.

S obzirom da se od korisnika najvećeg broja vaših programa na može očekivati da imaju poslavljenu vrednost **TZ** niti da im se uputstva komplikuju ovakvim objašnjenjima, posloji jednostavan način da tačku na kojoj se program izvodi učinite "vremenskim centrom sveta". Dovoljno je da negde na početku programa upišete

timezone = 0;
dayiight = 0;

i nakon toga neće biti nikakvog podešavanja vremena.

Ako vam ponekad u programu zatreba samo ascii zapis trenutnog datuma ili vremena, možete najlaće upotrebiti i MSC neslandardni dodatak **_strdate** i **_strtime**, na sledeći način

char sd[9],st[9];
printf("Datum: %s Vreme: %s \n",
_strdate(sd),_strtime(st));

GDE JE PROGRAM?

Gotovo da nema tole „dužeg" programa kome ne treba neka konfiguraciona datoteka. Zahvaljujući jednostavnosti u preuzimanju parametara iz datoteke koju omogućavaju funkcije za formatirano čitanje sadržaja (**scanf** grupa funkcija) korišćenje konfiguracionih datoteka je izuzetno jednostavno i fleksibilno i zaista se ne može naći bolji način da se veći broj parametara prenese u program. O načinu na koji se funkcije iz grupe **scanf** mogu upotrebiti za ovu namenu je već bilo reči u ovoj seriji u decembru. S obzirom da je tema bila sama funkcija, lada sam govorio samo o principu – za konkretnu realizaciju je važan još jedan detalj.

Osnovno je pitanje – gde smestiti konfiguracionu datoteku? Ono je naročito važno ako je program iz klase „utility", koji se skoro nikada ne startuju iz tekućeg direktorija – alate redovno slažemo u neki odvojen direktorij i zatim ih PATH promenjivom činimo dostupnim sa bilo kog mesta na disku. Time otpada mogućnost da konfiguracionu datoteku upisujemo u tekući direktorij.

Jedino uvek poznato mesto (naravno, ako zanemarimo neku fiksnu odrednicu poput C:\ koja je praktično neupotrebljiva) je mesto na koma se nalazi sam program. Jedna od standardnih funkcija operativnog sistema bi trebalo da bude i ona koja dozvoljava svakom programu da sazna lačno kako se zove i na kom mestu na disku se nalazi. Kod MS DOS-a je ovo ubačeno dosta kasno (mislim od verzije 3.xx), pa je i način realizacija adekvatan – podatak je nekako uguran u **environmet** oblast. U ovom području se kao što je poznato nalazi niz c-ovskih stringova (0 kao terminator) u kojimi IME=PARAMETAR. Iza poslednjeg stringa ubačen je još jedan terminator 0, pa iza njega sledi pun naziv programa.

Na sreću, C inicijalizacioni kod svakog programa o ovome vodi računa pa nema potrebe da se zamaramo traženjem. Ako je naša **main** funkcija deklarisana kao

main(int argc, char *argv[])

onda je **argv[0]** parametar koji nam je potreban. Pošto je on uvek kompletan, sadrži i staru i naziv programa iz njega možemo izdvojiti direktorij u kome se program nalazi. Na taj deo pojom „nalepimo" naziv konfiguracione, help i sličnih datoteka i možemo biti sigurni da smo se obratili na prvo mesto.

Treba voditi računa o tome da je za svaki string na koji ukazuje niz pointera **argv** prostor dinamički alociran i da nije baš zgodno menjati njegov sadržaj jer nama mesta za nazive duže



od naziva samog programa. Najjednostavnije je odvojiti neku globalnu promenjivu, recimo **prgpath** u nju odmah na početku programa prebacili sadržaj **argv[0]** (**strcpy(_prgpath, argv[0])**) i polom koristili jednostavnu funkciju za formiranje naziva konfiguracione datoteke.

U ostatku programa kada nam zatreba dovoljno je recimo

f = fopen(cfgname("XDC.CFG")),

Funkcija **strrchr** je deo ANSI standarda – vraca pointer na poslednji pojavu traženog znaka u zadatom stringu ili NULL ako znaka u stringu uopšte nema. Pošto znaka \ neće biti jedino ako zaboravimo da prebacimo **argv[0]** u **_prgpath** nama potrebe da posebno ispitujemo tu situaciju.

```
#include <string.h>
char _prgpath[79];
char *cfgname(char *name)
{
    char *bs;
    bs = strrchr(_prgpath,'\\');
    strcpy(bs+1,name);
    return( _prgpath );
}
```

Kada smo već kod ovog razdvajanja staze od naziva programa – Microsoft C poseduje i dve neslandardne funkcije za ovu namenu (od verzije 6.0 sve neslandardne dodatke ćete lako prepoznati po uvodnom znaku _ u nazivu). Funkcija

void splitpath(char *path, char *drive, char *dir, char *fname, char *ext),

će sadržaj **path** stringa rastaviti na sastavne dalove koje će smestiti u odgovarajuće promenjive koje joj prosleđujemo (za dimenzije stringova treba koristiti konstante iz **stdlib.h** _MAX_DRIVE, _MAX_DIR, MAX_NAME i _MAX_EXT). Ukoliko neki od delova ne postoji u nazivu odgovarajući string će biti prazan. Funkcija sa potpuno istom deklaracijom – **_makepath** će obaviti inverzni proces sastavljajući punu stazu od zadanih delova.

Funkcija **_splitpath** ipak nije rešenje za sve situacije. Problem je u tome što svoj posao obavlja „mehanički" – prosto analizira zadati string. Ako joj recimo prosledite

C:\DATA

u stringu **fname** će se uvek naći **DATA** čak i ako se radi o direktoriju a ne nazivu datoteke

SEZAM

Pripremio Dejan Ristanović

7

BILTEN

JUL 1991

Juli je na Sezamu kao i u ostalim javnim glasilima protekao u znaku dramatičnih događaja u zemlji. Ti događaji su nešto pre toka „prepunili" konferenciju FORUM pa je u skladu sa nobeljanim scenarijem za tu priliku ona proglašena za FORUM 2 dok je novi FORUM počeo da prima poruke – za svega nekoliko dana jula sakupilo ih se preko 600!

Kada je reč o brojnosti, tokom jula smo „opaljuli" 12 122 poziva, komunicirali sa korisnicima 2 683 časova i 87 dana (obzirom da mesec ima 31 dana u proseku je bilo **napreklidno** zauzeto 2.8 modova na sto manje nego prethodnog meseca što verovatno počiva vremena godišnjih odmora) od čega je u *chatu* provedeno 880 časova (i 110 radnih dana (i radna noći). Korisnici su sa Sezama *download*-ovali 303 megabajta dalotaka dok je za isto vreme sistem poslalo 16 megabajta.

Što se stvari „koje ostaju" tiče, tokom aprila je u konferencije pristigio oko 2824 poruke (3 3 megabajta taksta za 50% više nego u junu) i 3 7 megabajta dalotaka (kojih u istom mesecima ima nesto manje nego obično, pošto je i manje značajnih događaja) prslizale i brojne nekompatibilne poruke, mahom „divaklan pre nos" informativnog tala koji se povodom situacije u Jugoslaviji vod o i vodi na inostranim komjuterskim mrežama.

Konferencija **RAČUNARI** se tokom jula, baš kao i naš časopis, bavila problemom zaranim zvanim kašnjanje. Obzirom da je zbog problema sa transportom kvalitetnog papira preko granih područja letnji dvobroj Računara" kasnio čitavih mesec dana, čitaoci su se na poslednjoj konferenciji kada će časopis izaći a onda i komentarisali takvu situaciju koja se na pamti još od vremena „Računara u vašoj kući". Neki su za nju imali razumevanja neki baš i nisu. U svakom slučaju za sama „Računara" bi bilo jako dobro da se sličn problemi u budnosti na ponove.

„Ki vač" za svojevrsan bum poruka n konferenciji **SEZAM** (naročito pred kraj jula) je program SOR. Se *zam Off line Reader* koj je kreirao *Zoran Živatic*. Program omogućava da pošto sle preneli *pad* sa novim porukama sve što vas zanima prenesete u po sebnu batu podataka da poruke čitate prateć lance d skusija da p'šete odgovore i najzad da automatski kreirate axec koj čala pr sledećem javljanju prosledi Sezamu. Posto je napisana uvodna verzija programa i posto se pokazalo da je intres za nju ogromau zapoćela je diskusija o mogućim poboljšaujima čitavog sis ma. Diskusija je trzuli rala najpre varzijom 0 1 SOR a a onda početkom avuausta i SOR-om 1 0 ko ji predstavlja prvu potpunu verziju. Diskusija na ovu temu u dan ma kada ovo pišemo i dalje traja pa se očekuja da će SOR uskoro poprimiti svoj konačnan oblik. Program sa naravno može slobodno *downloa*dovali sa Sezama.

U kotaranciji **KOMUNIKACIJE** diskutovalo se o instalaciji nekih nestandardnih modema na PC-ju i *Macu script* jeziku za *Telix* pradnostima i mana ma MNP pranosa podataka ARJ arhiveru koji ne žalosti zahteva previše memorija da bismo ga instalirali kao jedan od „zvaničnih" Sezam arhivara novoj graličkoj verzu DSZ a BBS ovima u nekim isločnim zemljama (npr Čehoslovačka). Što se „večih" sistema liče distukovali smo o slabiln *Bitnet* a/EARN a i njegovim nedavnim (zlo)poli.abama u politčke svihe prenošenju PD softvara z inostranstva preko DECnet a. *Downloa* du sa VAX a problemima koja ljuba rašiti da se u lako prenesenoj binarnoj dalateci ne bi pojavile <CR><LF> kombinacuji viška" i sličnim atvarima. Nesnmnjiva lema meseca bio je svakako program PGP koj omogućava šifrovano komun c anje pomoćn javne sire odnosno RSA algoritma. Tim čudom matematike omogncno je da svoj javni ključ distribuišete svakome ko je zainteresovan i to javnim kanalima a da onda on šifruje poruke koje cete jedak samo vi moći da deš.frujete. Zvuči kao čarolja al je moguće i sto je još lepše uz pomoć ovog programa postaje sasvim jednostavno. Planiramo prikaz n ste dnjih Računarima a sada čamo reći da je prvih da ne po pojavi ovog programa na Sezamu bilo mnogo šifrovanog komnnicirnja čak i kroz konferencije a verovatno i kroz privatnn postu.

U konferenciji **PC HARD** se ponovo raspalmsala diskusija o tvrdm d skovima i njihovom povezivanju. Radi se uglavnom o AT Bus i IDE diskovima njiho vom formatiranju i vezivanju na red. Ova tema se na kako ciklično javlja svakih par meseci bukne iznova. Druga tema o kojoj se dosta govorilo je odnos brzina računara i hpa brzine organizacije memonje koja je pokazala da jes uvek posloj dosta netazumevanja oko ovoga. Ako je ovo neka indikacija onda bi n. Računarima" trebalo opet progovoriti o ovoj problematici. Bilo je reči i o video karlama kao i kraća diskusija na temu kakav monitor treba knpiti. Osnovno obeležje konferencije je dosta velika zivost nprkos sezon godišnjeg odmora i tekućim događajima.

Što se konferencije **PC SOFT** liče dominirala je tema *ms dos* naročito po pitanju DOS-a 5 0. Reakcije su oprlične raznovrsnom podeljene – od izvešlaja o problem ma pri upotrebi nekih programa (npr. Turbo Debugger u 386 mogu ako sa koristi DOS HIGH) do pohvala nov m mogucnostma. Rekli o še da kori sn ci IBM kompatibilnih računara lesko i izdrzevaju ograničenja DOS a pa svakn novu verziju čekaju sa nestrpljanjem. Tema *windows* je bila nesto „m m ja" nego obično verovatno zbog novog DOS a a uglav nom se radilo o poznatim problemima fontovi progra mi koji nece da rada u prozoru itd. U okviru tema *jezi ci* se izmašu ostalog pricalo o alatima koji transliši misu izvorni kod u FORTRAN-u u ekvivalentan kod u C-u isprobavali smo jedau program za olki vauje har dveiskog baga na lutalovm 386 i 486 procesoruma (viša o tomu u Bajlovima lična purode") opšti zaključak je da 486 procesori nemaju ovaj bag kao i vecina 386SX procesora a da ga svi 386DX proceson imaju uključujući i 386 procesore proizvodača AMD). Pričalo se o još jeduom problemu koj je izgledao kao softvarski da bi se na kraju pokazalo da je hardverski program NG (*Norton Guide*) „odbija saradnju" sa uakim Hercules kompatibilnim video karticama jedini lek ja po svamu sudać zamena video kartice.

Juli je u **ATARI ST** konferenciji bilo u znaku arhiva. Pojavom ST ZIP-a i nove verzije arhivara ZOO se novim brzim i moćnim kompresionim algoritmima pojavila se potanja da li će LZH ostali osnovni ST arhi var. Takođe je bilo i dosta novog kvalitetuog PD sof tvera: stotinak kilobajta vesti iz ST-sveta a otvoren je i novi direktori pun GAMES u kome se sada nalazi de setak novih igara.

Tokom jula je eksperimentalno otvorena konferencija **AMIGA** posvećena isloimanom računaru. Obzirom da na Sezamu ima dosta korisnika *Amiga* prvi dani ove konferencija doneli su preko 100 interesan tnih poruka.

Godišnji odmori i gradanski rat su opustili **ORKU** – tokom jula bilo je jedva šezdesetak poruka. Bilo je uajviše reči o igrama, naročito *Elite Plus* i *Hitchhiker s* igrači ove druge su čak osuvali svoju grupu. Što se „ozbiljnih" stvari liče, izazen je program koji bi isključio PC jev zvučnik i naden jedan koj ne radi baš uvek. Oslavljen je program za emulaciju Amigu na PC ju a diskutovalo se i o kvarovima koje izazivaju tropske vlucine. *Nenad Mijailović (nesavic)* je ostavio svoj program za kodiranje/dekodiranje na testiranje i pomio uglavnom veoma povoljne kritike.

Juli u **FORUM**u obeležen je posleđnim danima „rata" u Sloveniji početkom rata (bez navodnika) u Hrvatskoj. Sezamovci su se nkljnili n. nformativnu tal po svetskim računarskim mrežama neki direktno biju ci bilke a neki prenoseći odjeka na Sezam. Na sa mom Sezamu nije toliko izražena podela na one koji su za rat i one koji su protiv rata (uglavnom su svi barem deklarativno protiv rata) koliko je izražena podela na one koji smatraju da se do mira (pa i kasnjeg prosperiteta) može doć tak kada se stvari prvo počis te kod svoje kuće i onih koji smatraju da pak prvo treba dobiti rat a za ostalo će biti vremena. Posto je FORUM dosligao svoj limit od 5000 poruka preimenovali smo ga u FORUM 2 otvorili novu konferenciju koja se uglavnom puni" porukama političke sadržine uz neško lračeva i neizbeznu temu *devojke*.

Da li zbog tnlobna situacije il zbog jednomesečnog odsustva moderatora n konferenciji **VICEVI** bilo je teško mirnije nego obično. Najvše pričoga bilo je na temu *mih* (Minja i Hasol a ni političkih viceva nije nedostajalo. Glasanjem je narevno izabran najbolji vic jnna koji je predugacak da bismo ga ovde delajino citirali.

KAKO POSTATI ČLAN

Postupak učlanjenje počinje zvanjem Sezama. Na kom pozive broja 646-681 i uspostavljanja veze (parametri 2400N8) pozdravne maske dobićete pitanje

Username

Na njega treba odgovoriti sa NEW (new korisnik) nakon čega vas Sezam već ima prvog pnjave. Prvi podatak koji treba da date je vaše pravo ime i prezime o potom birate pseudonim pod kojim cete kasnije sistem. Pseudonim je bilo kakvi niz do 18 znakova koja sadrži sama mala slova i znak (tačka) i obično se bira tako da asocira na vaše ime. Sistem će na osnovu vašeg pravog i mena predložiti pseudonim koji možete ali i ne morate prihvatiti. Pre zbora pseudonema dobro razmislite jer on postaje vaše lično ime na Sezamu i ne možete ga više promeniti.

Sledeći korak je zbor lozinka (password) kojim se vaš pseudonim štit prijavljivanje sistemu je moguće samo ako poznajete lozinku koja je vezane uz vaš pseudonim.

Nakon ovih osnovnih podataka sledi izbor nekoliko raznih parametara (objašnjenja ćete dobiti na ekranu) a potom unos ostalih ličnih podataka – adrese telefona i podataka o ojemu koji koristite za zvanje Sezama. Ovim je prva faza gotova.

Posebno obratite pažnju da adresu unesete korektno – upravna adresa je jedin način da Uprava sačuvati dalju kontakt se vama kako bi se da kraja obavio postao vašeg učlanjenje

Sledećeg radnog dana će vam biti poslate poane u kome cete primati i popunjena uplatnico. Na vama je da naberete penod na koj se učlanjujete i na osnovu toga ne uplatnicu upišete unos pratplate

za 3 meseca	400 din
za 6 meseci	700 din
za godinu dana	1100 din

Nakon izvršenm uplate fotokopiju potvrde o uplati pošaljite na naznačenu adresu Uprave Sezama telo poštom ili (daleko dalje) faksom. Po prjemu ove potvrde Uprava da vam istog radnog dana odobrib pristup sistemu čime postaje punopravan član Sezama. Istog dana će na vašu adresu bili poslati i Uputstvo za korišćenje sistema

Ukoliko pri prvom prijavljivanju imate problema s na stavšta popunjavanje podataka korektno nije potrebno da svete javljate i pnjave jer će se poslovim menom Sačekajte jedan dan pa ponovite ponovo – adm nistraton sistema kod detektovanja neprpunih prijava intervenisu daju vam mogućnost da pri sledećem javljanju ponovo popunite prijavu ili vam ostavljaju ličnu poruku kojom vas obaveštavaju šta treba dalje da učinite

Ukoliko ste se ipak prijavili pod više različitih pseudonima, mapte ne ama da Uprava sama mora da ovrši izbor koju od vaših prijava će uzeti u daju obradu. Kriterijum je najčešće kompletnim podataka. U slučaju više korektnih prijava uzima se n slizu same poslednja. U svakom slučaju ha uplatnico će biti upisan pseudonim na koje se uplata odnosi i samo njega možete koristiti.

Ukoliko želite pre pravljanje da razgledate sistem na pdanje Username odgovorite sa DEMO Naći ćete se n demonstracionom sistemu iste u kome možete pregledati kompletnu strukturu sistema kaja je identične onoj pravoj. Ako DEMO pozivate iz je ograničen a vremena za jednu sesiju je 5 minuta.

SezamFile 1

Priprema: Jovan Bulajić

Nikada nam nije bila ambicija da stvorimo BBS kome će se ljudi javljati samo da bi sa njega skupljali javni softver. Stoga smo ovaj segment sistema u početku namerno stavili u drugi plan ograničivši ga samo na mali broj pažljivo probranih programa. Sada, kad su se konferencije razvile a Sezam dobio upravo onu fizionomiju koju smo i hteli da mu damo nema nikakvog razloga da na pojačamo i njegovu softversku bazu. Poslednjih meseci broj datoteka u Sezamovim direktorijumima je naglo počeo da raste. Ovu rubriku otvaramo da bismo vam olakšali praćenje onoga što se u njima dešava

JAVNI SOFTVER

Na Sezamu se mogu naći isključivo programi čija je kopiranje i slobodna distribucija ne samo dozvoljena nego čak i poželjna. Tu mislimo na programe u javnom vlasništvu (*public domain, sharewara, freeware i user suported*) i druge oblike takvog softvera. Autori ovakvih programa dozvoljavaju njihovu slobodnu distribuciju pod uslovom da se za to ne traži nikakva (ili samo minimalna) nadoknada od korisnika. Ukoliko sle zadovoljni programom ti imate nekakve koristi od njega možete autoru poslati neku malu nadoknadu. U svetu (a ima primera i kod nas) postoje kompanije koje se bave distribucijom izabranog *public domain* softvera na disketama pa je to često jedini put za one koji nemaju modem da dođu do jeftinog i kvalitetnog softvera.

Od sada ćemo, zbog kraćeg i jasnijeg izražavanja, sve ovakve programe nazivati jednim zajedničkim imenom – *public domain* ili PD softver.

Tabela 1

ATARIST	<DIR>
COM	<DIR>
IBMPC	<DIR>
NOVOSTI	<DIR>
PITALICE	<DIR>
RSOFT	<DIR>
UNIX	<DIR>

Osnovni direktorijumi na Sezamu

Tabela 2

Grane direktorijuma	Broj datoteka	Ukupna dužina
IBMPC	21	2 153 426
ATARIST	70	2 010 643
Ostalo (UNIX RSOFT NOVOSTI)	8	46 600
Ukupno	99	4 210,669

Nove datoteke u julu

Ukoliko je neki program *shareware*, takođe je dozvoljeno njegovo kopiranje i ograničeno korišćenje. Ograničenost se ogleda u tome što program možete slobodno koristiti neko određeno vreme (npr. 90 dana) a posle toga ga ukoliko ga želite koristiti i dalje morate registrovati tj. uplatiti autoru određen iznos. Zavisno od programa, registracijom stičete razne pogodnosti – pravo na nove verzije, dobijate odštampanu dokumentaciju (priručnik za rad) drugu (ne-*shareware*) verziju programa ili nešto slično. Ponekad je *shareware* verzija samo skraćeni prikaz mogućnosti prave komercijalne verzije pa registracijom dobijate i nju.

Cilj ove rubrike je, između ostalog, i da preko opisa programa sa Sezama popularnše PD softver. Za nezavisne programere, to je najbolji (a često i jedini) način da omoguće distribuciju svojih programa. Zahvaljujući razgranatim računarskim komunikacijama, brojni PD programi su postali dostupni praktično svakome u kratkom vremenskom roku. Dovoljno je program ostaviti na nekoliko većih sistema (npr. na BIX i na Sezam...) a odatle će se vrlo brzo, ukoliko kvalitetom to zaslužuje, proširiti do svih krajeva računarskog sveta.

SOFTVER NA SEZAMU

Datoteka su organizovane na način na koji ste navikli u DOS u, tj. podeljene su u direktorijume. Ukoliko sle novi na Sezamu (ili to planirate da poslanate uskoro) a nestrpljivi ste da vidite čega sve ima po direktorijumima evo kratkog podsetnika komandi namenjanim radu sa datotekama. Između direktorijuma se kretaje komandom od <**ime direktorijuma**> (kao što ste navikli u DOS u), a tu je i **ncd** <**početak imena**> za ubrzano kretanje (kao što ste navikli ako konstile *Norton Utilites*). Spisak datoteka dobijate sa **dir** (i ovo vam je poznato sem ako se nisle rodili sa mišem u ruci) a datoteku prenosite sa **dow** (ili **download**).

Specifičnosi Sezama su i datoteke vezane uz poruku u neko od konferencija. U početku rada je namera bila da ta mogućnost služi slanju kraćih datoteka koja bi bile vezane za odrađenu diskusiju ali se vremenom došlo do sistema „Evo programa za... mozda nekom za treba". Tako su datoteke uz poruke pretvorene praktično u bajer za smašlanje zanimljivog softvera, dok ga neko od nadležnih (ili moderator) ne pregleda i odredi mu mesto u nekom direktorijumu.

Uz strogo poštovanje autorskih prava druga stvar o kojoj se vrlo vodi računa je provera programa od eventualne zaraženosti virusima. Svaka datoteka se pre stavljanja u neki od direktorijuma proverava na prisustvo virusa uz pomoc poznatih PD (shareware) virus skenera. Od njegovog osnivanja na Sezamu nije bilo ni jednog slučaja da sa virus pojavio u ostvaljenim programima, a hrudimo se da tako i ostane.

Spisak osnovnih direktorijuma možete videti u labeli 1. IBMPC i ATARIST imaju razgranatu mrežu poddirektorijuma. Nas će interesovati samo IBMPC deo (koji uključuje i COM direktorijum). Trenutno ima devet (uz COM deset) direktorijuma u kojima se nalazi softver za PC – ARCHIVER, DEMO, DRIVERS, FUN, GRAPHICS, MISC, UTILITY, VIRUS i YU. Iz naziva se naslućuje koja vrsta programa se nalazi u njima, ali ćemo detaljniji opis oslaviti za sledeći broj, pošto su u planu izmene u IBMPC stablu koje će doneti nekoliko novih direktorijuma.

Trenutno se na Sezamu nalazi oko 140 raznih programa za PC ukupne dužine oko 10 MB. Sva daljeleke su u arhiviranom obliku i to spakovane PKZIP arhiverom koji je (još uvek) zvanični arhiver Sezama a neke se nalaze i u obliku samoraspakivajućih arhiva. Možda 10 M nakome ne zvuči mnogo, ali su to zaista pažljivo probrani programi čiji je broj počeo ubrzano da se povecava tek u zadnja dva-tri meseca. Tako je samo u julu pristigło više od 2 MB novih daloteka, a slična tendencija se očekuje i u avgustu.

U sledecim nastavcima ovog biltena predstavljaćemo najzanimljivije od novog PD softvera koji pristiže na Sezam. U početku ćemo ići malo ubrzanim tempom, pošto treba predstaviti najbolje od softvera koji je već na Sezamu, a o njemu nije pisano u „Računarima" „Računar" su u poslednje vreme imali nekoliko članaka o PD softveru (4DOS komunikacioni programi, arhiveri) ali pošto nisu svi programi koji zaslužuju da budu spomenuti, pikladni za „velika" članke, to će njima biti bitl posvećena pažnja u ovoj rubrici.

DOGODILO SE U JULU

U julu se na Sezamu našla 21 nova datoteka programa za PC ukupne dužine preko 2 MB, kao što se može videti iz tabele 2. Najzapaženiji novitati su nova verzija McAfee porodice antivirusnih programa, program za šifrovanje i ARJ arhiver (koji je dogurao do verzije 2.2) ali je bez sumnje dogadaj meseca pojava *Sezam Off line Readera* o kome ce sigurno biti reči u nekim drugim člancima u „Računarima".

Top listu najtraženijih programa (sa najvećim brojem *downloada*) možete videti u tabeli 3. Poltudićemo se da programi koje budemo predstavljali u narednim brojevima budu oni sa vrhova top lista. Da bisle imali osećaj o kojim se ciframa radi, reći ćemo da je najatraktivniju datoteku u julu (sor.zip) sa Sezama uzalo čak oko 130 korisnika za period od samo oko dva nedelje (pošlo se sor pojavio sredinom meseca), a *scan80* je samo malo zaostao.

I jedna napomena članovima Sezama vezana za top listu – neke daloteke nećete naći pod jim imanima u direktorijumima pošto su zbog svoje duzine podeljene na nekoliko delova. Tako npr. *Graphic Workshop* (gws52.zip) mozete naći u datotekama gws521.zip i gws522.zip a bench56.zip kao bench561.zip i bench562.zip.

Toliko za početak i spremite sa na dosta PD softvera koji vas očekuje u sledecim brojevima. Najbolje ćele se pripremiti nabavkom modema (ako ga već nemate) i učlanjenjem na Sezam (ako već niste član).

TOP 10 Datoteka – JUL 1991

	direktorijum	datoteka		opis
1	\ COM	sor	zip	Sezam OFF LINE READER (Verzija 0.1)
2	\ IBMPC \ VIRUS	scanv80	zip	McAfee virus detektor
3	\ IBMPC \ VIRUS	clean80	zip	Uklanja viruse koje Scan nađe
4	\ IBMPC \ VIRUS	vshld80	zip	Rezidentni Scan
5	\ COM	ats	zip	Script za automatizovanje rada na Sezamu
6	\ IBMPC \ FUN	dash	zip	Boulder dash
7	\ COM	pgp10	zip	RSA šifre – neprobojno šifrovano komuniciranje
8	\ COM	dsz0613	zip	ZMODEM protokol (DSZ verzija 13.06.91)
9	\ IBMPC \ GRAPHICS	gws52	zip	Graphic Workshop – rad sa mnogim grafickim formatima
10	\ IBMPC \ MISC	bench56	zip	PC Magazine benchmark (brzinski) testovi

Najtraženije datoteke u julu

RAD SA FONTOVIMA

Najveći deo komunikacije sa štampačem svodi se, sasvim prirodno, na ispisivanje teksta. Zato cemo ovoga meseca upoznati komande PCL-a koje operišu sa fontovima.

Dejan Ristanović

Za razliku od *LaserJet-a* II u čiji su ROM bila ugrađena svega tri fonta (*Courier 10 Pitch, Courier Bold 10 Pitch* i *Line Printer 16 66 Pitch*) *LaserJet* III je daleko bolje opremljen. Štampac naime i bez dodatne opreme omogućava ispis teksta u nekom od sedam bit-mapiranih ili dva skalabilna fonta. Obzirom da smo stigli do veoma konkretna diskusije nace skodili da se pre nego što prademe na komande podselimo razlike između ove dve vrste fontova.

JOŠ JEDANPUT IZ POČETKA

Bit mapirani font je otprilike, ono što zamišljamo kada se pomene definicija slova – štampacu je na neki način „saopštena" matrica (rezolucija 300 tačaka po inču) tacaka i praznina koje čine oblik slove i na njemu je semo da tu matricu. Jakvu kakva je prenese na papir. To praktično znaci da jedan bit mapirani font predstavlja jednu veličinu slova – ako nam je potrebna druga veličina moramo poslati (ili imati u ROM-u) drugi font. To takođe znaci da se svako slovo može pregledati na ekranu i uz pomoć nekog uslovno rečeno jednostavnog programa popraviti i ponovo smestiti na disk odnosno u dalom trenutku poslati štampacu. Dodavanje YU slova je kada se radi o bit mapiranim fontovima, dosadan ali u potpunosti definisan posao koji se može završiti za određeno vreme koje zavisi od vaše dizajnerske sposobnosti kvalitela *font editora* koji koristite i, naročito vašeg iskustva u ovakvim poslovima.

Skalabilni (navikli smo na termin *skalirani*, all on nije dovoljno precizan) font je nešto sasvim drugo – slova su definisana kao niz pravih i krivih linija, dakle vektorski. To znači da se iz jedne jedine definicije mogu na zahtav korisnika generisati sve potrebne veličine slova što je nesumnjiva prednost u odnosu na bit mapirane fontove kako sa aspekta fleksibilnosti tako i sa aspekta zauzeća memorije i brzine komunikacije. Na žalost sve ima svoju cenu – skalabilni font sa sastoji od nekih „nerazumljivih" krivulja pa ga na možete tek tako ispravljati i prilagođavati svojim potrebama. Nećemo da kažemo da je takvo ispravljanje **nemoguće** – samo je znatno laže i zahteva posedovanje ne samo znanja i dizajnerskog dara nego i odgovarajucih (potencijalno vrlo slozenih) softverskih alatki. Obzirom da u skalabilnim fontovima nema svih YU slova, njihova upotrebljivost za domacag korisnika laserskog štampača je više nego diskutabilna, tako da ćele u praksi *LaserJet III* ugiavnom koristiti na bazi bit mapiranih fontova. U okviru ove serije tekstova cemo medutim paralalno obradit i jedne i druge.

Što se izvorišta fontova tice, razlikujemo ugradene fontove (fontova iz kartridža i takozvane *soft* fontove odnosno datoteke u koje su upisani oblici slova koji se pre štampanja prosleđuju laserskom štampacu i upisuju u njegov RAM. Dok su oblici slova u ROM-u i kartridzima nepromenljivi *soft* fontove pre slanja štampaču mozemo pro izvoljno prepravljati (npr. pomoću *FontGen-a* ili *Soft Craft Font Editor-a*) i tako zamenit nekе manje potrebna slova našim čćđž i drugim znacima. Konščenje *soft* fontova, osim loga, ne zahteva nabavku posebnog hardvera (ROM odnosno EPROM moduli sa fontovima nisu ni malo jeftinnije kada bi svvrlo široko (*Venture, Pagemaker WordPerfect*...) a mogucnosi kombinovanja potpuna. Loša strana ovih fontova je što zauzimaju RAM štampača (ukoliko neproširenom *LaserJet-u* III koji ima megabajt memorije pošaljete font veličine 72 pt za druge fontove grafiku i sam tekst će ostati jako malo prostora!) i što njihovo ucitavanje zahteva određeno vreme.

ČIME RASPOLAŽEMO?

Upoznajmo pre svega bit mapirana fontove ugradana u ROM *LaserJet III*. Radi se o *Courier* fontovima neproporcionalnom pismu (sva slova jednako široka za razliku na primer od teksta koji čitate i svih ostalih knjiga) koja veoma podseća na pisaću mašinu. Osim standardnih

RAD SA FONTOVIMA

<SI>	0F	Aktivira primarni set znakova
<SO>	0E	Aktivira sekundarni set znakova
<Esc> (** **	1B 28 ** **	Potreban set znakova sa oznakom ** ** (brojka i slovo) kao primarni
<Esc>)** **	1B 29 ** **	Potreban set znakova sa oznakom ** ** (brojka i slovo) kao sekundarni
<Esc> (s1P	1B 28 73 31 50	Potreban proporcionalan font kao primarni
<Esc> (s0P	1B 28 73 30 50	Potreban neproporcionalen font kao primarni
<Esc>)s1P	1B 29 73 31 50	Potreban proporcionalan font kao sekundarni
<Esc>)s0P	1B 29 73 30 50	Potreban neproporcionalan font kao sekundarni
<Esc> (s**H	1B 28 73 ** 48	Potreban primarni font čiji su znaci široki ** decimalnih tačaka
<Esc>)s**H	1B 29 73 ** 48	Potreban sekundarni font čiji su znaci široki ** decimalnih tačaka
<Esc> (s**V	1B 28 73 ** 56	Potreban primarni font čiji su znaci visoki ** decimalnih tačaka.
<Esc>)s**V	1B 29 73 **-56	Potreban sekundarni font čiji su znaci visoki ** decimalnih tačaka.
<Esc> (s**S	1B 28 73 ** 53	Potreban font čiji je stil ** (1 za kurziv) kao primarni
<Esc>)s**S	1B 29 73 ** 53	Potreban font čiji je stil ** (1 za kurziv) kao sekundarni
<Esc> (s**a	1B 28 73 ** 42	Potreban primarni font čije je zacrnjenje ** Argument je između -7 i +7
<Esc>)s**B	1B 29 73 ** 42	Potreban sekundarni font čije je zacrnjenje ** Argument je između -7 i +7
<Esc> (s**T	1B 28 73 ** 54	Potreban primarni font iz familije **
<Esc>)s**T	1B 29 73 ** 54	Potreban sekundarni font iz familje **
<Esc> (3@	1B 28 33 40	Bira primarni font prema stanju na panelu
<Esc>)3@	1B 29 33 40	Bira sekundarni font prema stanju na panelu
<Esc> *c**D	1B 2A 63 ** 44	Radiće se sa identifikacionim brojem ** Broj je između 0 i 32767
<Esc> *c0F	1B 2A 63 30 46	Briše sve soft fontove i njima dodeljene id brojeve kao i id brojeve svih fontova iz ROM-a. Pri brisanju se ne obraća pažnja na to da li je font (id broj) označen kao stalan ili kao privremen
<Esc> *c1F	1B 2A 63 31 34	Briše sve privremene fontove kao i privremene id brojeve dodaljene ROM fontovima
<Esc> *c2F	1B 2A 63 32 34	Briše trenutno izabrani font.
<Esc> *c3F	1B 2A 63 33 34	Briše poslednji karakter koji je definisan u fontu koji je trenutno aktivan
<Esc> *c4F	1B 2A 63 34 34	Proglašava izabrani font privremenim
<Esc> *c5F	1B 2A 63 35 34	Proglašava izabrani font permanentnim
<Esc> *c6F	1B 2A 63 36 34	Dodeljuje tekući id broj tekućem fontu
<Esc> (**X	1B 28 ** 58	Aktivira font čiji je id broj ** i proglašave ga za primarni
<Esc>)**X	1B 29 ** 58	Aktivira font čiji je id broj ** i proglasava ga za sekundarni

Slika 6 Naredba za rad sa fontovima

Slika 1 Oznaka akupova znakova

Set simbola	Kod	Set simbola	Kod
Osnobitni		**Sednobitni**	
Roman-8	8U	ASCII US	0K
Xana-8	8K	ASCII DS	0U
Math-8	8M	Chinese IRO	2K
ANSI/Windows	9U	Norvegian 1&0 v1	0D
PC-8	10U	Norvegian 1&0 v2	1D
Dekmarh PC-7	11U	Engilsh	1A
Cyrillic 1&08856/5	0R	Engilsh 2	2D
PC Cyrillic	2R	Franc0 IRO	0F
		Franc0 IRO 2	1F
ne-alfabetski		Germac NF	0G
		German IRO	1G
Line Draw	0B	IRV	2D
Math Symbols	0M	Italian	1I
Maths	9U	Japanese	1J
Math5	1S	Latin IRO	0N
Math7	0K	Portuguese IRO/IRM	5S
MlPvnt	15U	Portuguem IRO	4S
PiFonts	2U	Roman Extension	0X
ROM-A	9S	Spanish NF	1S
ROM-B	1U	Spanic0 1&0/IRM	0S
Mar 1 of 8	0Y	1&0	2S
ZAR/UPC	8Y	Swedin0	1E
		Swedish/Finnish	0S
		US Legal	1U

Slika 2 Međuredni razmak

```
   10 ESC-CHR$(27)
   20 LPRINT ESC;"&l12c"          'line spacing 12/48   (10 tačaka)
   30 LPRINT ESC;"(8Y"            'visina fonta 8 tačaka
   40 LPRINT "Visina  8 tačaka"
   50 LPRINT ESC;"(s10V"          'visina fonta 10 tačaka
   60 LPRINT "Visina 10 tačaka"
   70 LPRINT ESC;"(s12V"          'visina fonta 12 tačaka
   80 LPRINT "Visina 11 tačaka"
   90 LPRINT CHR$(12)             'Form Feed
  100 END
```

Program ilustruje red sa međurednim razmakom i različitim visinama slova

Slika 3 Oznake familija slova

Familija slova	Kod	Familija slova	Kod
Line Printer	0	Ospf Humanist (Optima)	17
Pica	1	Garamond	18
Elite	2	Cooper	19
Courier	3	Cormeat	20
Helvetica (Swiss)	4	Bernadmy	21
Times Roman (Dutch)	5	Bodoni	22
Gothic	6	Century Schoolbook	23
Script	7	Univercity Roman	24
Prestige	8	Avant Garde Gothic	25
Caslon	9	Marlinn	27
Orator	10	Bitstream Charter	28
Presentation	11	Clarinet Black	29
Helvetica Condensed	12	Helliard	30
Future	16	Future Book	110
Palatine (Ospf Call)	16	Future Light	140
Souvenir	16	Helvetica Light	145

Slika 5 Izbor fontova

```
   10 REM
   20 REM Fontovi ugradeni u LaserJet III
   30 REM          (PC Karakter set)
   40 REM
   50 ESC$=CHR$(27)
   60 REM Courier 10 pitch normal
   70 LPRINT ESC$;"(10U"
   80 LPRINT ESC$;"(s0p10h12v0s0b3T";
   90 LPRINT "Courier 10 pitch normal"
  100 REM Courier 10 pitch bold
  110 LPRINT ESC$;"(10U";
  120 LPRINT ESC$;"(s0p10h12v3s3b3T";
  130 LPRINT "Courier 10 pitch bold"
  140 REM Courier 10 pitch italic
  150 LPRINT ESC$;"(10U";
  160 LPRINT ESC$;"(s1p10h12v0s1b3T";
  170 LPRINT "Courier 10 pitch italic"
  180 REM Courier 12 pitch normal
  190 LPRINT ESC$;"(10U";
  200 LPRINT ESC$;"(s0p12h10v0s0b3T";
  210 LPRINT "Courier 12 pitch normal"
  220 REM Courier 12 pitch bold
  230 LPRINT ESC$;"(10U";
  240 LPRINT ESC$;"(s0p12h10v3s3b3T";
  250 LPRINT "Courier 12 pitch bold"
  260 REM Courier 12 pitch italic
  270 LPRINT ESC$;"(10U";
  280 LPRINT ESC$;"(s0p12h10v1s0b3T";
  290 LPRINT "Courier 12 pitch italic"
  300 REM Line Printer normal
  310 LPRINT ESC$;"(10U";
  320 LPRINT ESC$;"(s0p16.66h5.5v0s0b0T";
  330 LPRINT "Line printer normal"
  340 LPRINT CHR$(12);
  350 END
```

Program ilustruje izbor osnovnih bit mapiranih fontova iz ROM-a *LaserJet*-a III

Oznaka stilova

Stil	Kod
Normalna (Upright)	0
Italic	1
Condensed	4
Condensed Italic	5
Extra Condensed	6
Expanded	24
Outline	32
Inline	64
Shadow	128
Outline Shadow	160

slova (u HP-ovoj terminologiji *medium upright*) na raspolaganju su polucrna slova (**bold**) i kurziv (*italic*) ali **ne** i polucrni kurziv. Raspoložive su samo dve veličine: 10 tačaka (pogodno za listinge pisma i tome slično) i 12 tačaka (što nije pogodno golovo ni za sta, iako ima i nekih kratkih pisma). Oni koji žele da pišu programe kompatibilne sa prethodnim *Hewlett Packard*-ovim štampačima treba da znaju da je *LaserJet II* u ROM-u imao samo *Courier* od 12 tačaka i to standardna i polucrna slova kurziv nije postojao baš kao ni *Courier 10 pt*. Iz razloga kompatibilnosti čak uključeni *LaserJet III* aktivira font *Courier 12 pt* koji ostaje izabran dok se komandom ne naloži drugačije.

Na raspolaganju je i *Lineprinter* font veličine 8.5 tačaka. Radi se o jako sitnim slovima pogodnim za neke listinge ili složene labele pri čemu ROM sadrži samo normalna slova, polucrna i kurzivna slova ovde ne bi ni imali mnogo smisla, pa nisu ni podržana. Isti font sadržao je i prethodni *LaserJet II* uz identična ograničenja.

Što se skalabilnih fontova tiče, na raspolaganju su familije *CG Times* i *Univers* čak visi *Helvetice*. Da bi stvari bile još lepša na raspolaganju su kako normalna (*medium upright*) tako i *bold*, *italic* pa čak i *bold italic* slova. Pitanja veličine se ovde ne postavlja pošto *LaserJet III* automatski skalira svaki od fontova na veličinu koja je potrebna korisniku u rasponu 0.25-999.75 tačaka. U praksi će retko koristiti fontova između 7 (eventualno 6) i 32 (eventualno 72) tačaka.

Učitavanje fontova zahteva poznavanje određenog skupa komandi i zadavanje brojnih parametara kojima ćemo se baviti sledeće meseca. Samo učitavanje naravno, nije dovoljno – treba upoznati komande kojima se neki od fontova aktivira ili

Na prvi pogled bi bilo najjednostavnije kada bi svaki font imao broj i kada bi postojala komanda „od sada slampaj fontom broj NN". Ovakvo komandovanje, međutim, *Hewlett Packard* nije usvojio iz razloga koje ćemo se kasnije baviti – kod prvog *LaserJet*-a fontovima uopšte nisu mogli da se dodeljuju identifikacioni brojevi, dok je na *LaserJet+*, *LaserJet II* i *LaserJet III* dodeljivanje brojeva moguće, ali i dalje ne predstavlja primarni metod upravljanja ispisom.

Kako se onda aktiviraju fontovi? Slanjem opisa, kažemo štampaču nešto poput „treba mi *Times*, 10 pt, PC set karaktera, malo crnje od normalnog". Štampač pronađe ono što najviše liči na naš opis i štampa u tom fontu. Do nedavno je ovakvo definisanje bilo sasvim besmisleno – u ROM-u svega tri fonta, 512 K RAM-a za definicije, sva skupa jedva deselak fontova, a mi šaljemo opis kao da se bira jedan od hiljadu! Otkako je *LaserJet III* uvao skalabilne fontove, ovakvo definisanje je dobilo pun značaj, premda cele ga i dalje radeći sa bit mapiranim fontovima slabo koristili. Moramo mi, međutim, posvetiti dosta prostora jer se u pojedinim situacijama bez njega jednostavno ne može.

PARAMETRI FONTOVA

Osnovna karakteristike prema kojima se bira font su

Set simbola određuje koji ASCII kod proizvodi koji oblik. Što se kodova 32-126 tice, stvari je prilično standardizovana – zna se da je 33 blanko, 48 nula, 65 veliko A, 97 malo a i tome slično, zaljubljenici u (već pritično zaslarali) sedmobitni YUSCII unekoliko menjaju ovaj raspored (kod 91 je, na primer, nije znak [nego slovo Š), ali se ta promena obično ne registruje u opisu fonta. Među kodovima 128-255 (kod PCL-a se obično govori o kodovima 160-255, pošto se u rasponu 128-159 na *LaserJet*-e ne mogu naći znakovi, kod *LaserJet*-a II i III ovo ograničenje ne postoji, ali se segmenti i dalje u praksi retko koristi, više iz navike nego iz potrebe za kompatibilnošću) vlada pravi haos. PC set, *Ventura* set, *Symbol* set, *Math* set – *Hewlett Packard* je propisao oznake selova znakova sa slike 1 (posloje i neke dodatne oznake koje su, međutim, toliko retke da na njih nećemo trošiti prostor), pri čemu se najčešće koriste PC 8 (IBM PC set znakova) i *Roman 8*. Kada budemo opisivali zaglavlje fonta, videćemo da bilo koja kombinacija cifra slovo može da predstavlja set znakova, nema, dakle, nikakva prepreke da formiramo fontove zvane 8J (8Y je zauzeto) koji bi predstavljali *Roman-8* set znakova sa YU slovima. Ovako nešto, međutim, malo ko radi.

Set znakova se određuje komandom **Esc** (oznaka primetimo da formal ove komande u suštini odudara od standardnog oblika PCL sekvenci (*ESC grupa argument znak*) ali je zahtevom da se ime seta znakova sastoji od brojke i slova formalna kompatibilnost sačuvana. Na primer, sa **Esc** (10U aktiviramo PC set znakova.

Proporcionalnost određuje da li je font proporcionalan ili nije – kod neproporcionalnog fonta sva su slova jednako široka, dok je kod proporcionalnog fonta W znatno šire nego I. Proporcionalni font se zahteva sa **Esc** (s1P a neproporcionalni sa **Esc** (s0P

Svi bit mapirani fontovi ugrađeni u ROM *LaserJet-a III* su nepropor cionalni ali su zato skalabilni fontovi kao i najveći deo *soft* fontova pro porcionalni

Širina slova (pitch) kod proporcionalnih fontova *pitch* označava ši rinu svakog karaktera izraženu u broju znakova po inču tipične vredno sti su 10, 12 i 16 66 CPI Osnovni font iz ROM-a *LaserJet* a i većine kompatibilnih laserskih štampača je neproporcionalan, dok je širina slo va 10 CPI

Kod proporcionalnih fontova svako slovo je različito široko što znači da je sam smisao pojma *pitch* promenjen usvojeno je da umesto širine slova *pitch* predstavlja širinu blanko simbola (ASCII kod 32) u fontu

Širinu slova zahtevanog fonta definišemo sekvencom **Esc (s#H** gde smo sa # označili širinu broj može da bude ceo ili decimalan na primer **Esc** (s16 66H Komandom **Esc &k#S** lakođe utičemo na širinu slova ali sa radi o „nasilnoj" promeni širine blanko simbola u cilju uravnavanje desne ivice teksta Ovu komandu ćemo upoznati nešto docnije

Visina slova se kao i širina izražava u decimalnim tačkama i pred stavlja parametar lonta za razliku od nešto ranije pomenutog *line spa cing*-a (komande **Esc &l#C)** koji označava razmak između redova i **obuhvata** i visinu slova Program sa slike 2 definiše fiksan razmak izmeđ u redova od 12/46 inča (nova jedinica pozdravite *Hewlett Packard*) odnosno 18 decimalnih tačaka (decimalna tačka je sečamo se 1/72 nča) a zatim bira fontova visine 8 10 i 12 tačaka u zavisnosti od kar tridža i učitanih *soft* fontova rezultat izvršenja programa može da bude različiti

Visina slova se kao što vidimo bira sekvencom **Esc &l#V**, gde je „larabom" označen argument izražen u decimalnim tačkama

Stil Određuje stil ispisa Nekada su postojali samo stilovi *upright* i *italic* ali je počevši od *LaserJet IID*, *Hewlett Packard* dodao nove stilove koje opisuje tabela 3 Odgovarajuća komanda je **Esc (s#S** gde umesto # treba staviti oznaku stila u skladu sa tabelom U praksi se međutim golovo isključivo koristi **Esc (s1S** (što aktivira kurziv) i **Esc (s0S** za povratak na uspravna slova

Kada budemo govorili o zaglavlju fonta videćemo da je takođe po cevši od *LaserJet IID*, *Hewlett Packard* proširo polje *style* na dva bajta (do tada je mogući raspon vrednosti bio 0 255) što bi u budućnosti trebalo da obezbedi dodatne stilove

Zacrnjenje (Boldface, Stroke Weight) obezbeđuje ispisivanje važ nijeg teksta polucrnim (**bold**) slovima Autori PCL-a su odredili da zacr njenje bude u intervalu [7 7] pri čemu 0 označava normalna slova pozitivni brojevi zatamnjenje a negativni svetlije (*light*) fontove Praksa nije sledila teoriju – na žrtvištu cete naći fontove čije je zacrnjenje 0 +3 (polucrno) i praktično nišla više

Izbor fonta odgovarajućeg zacrnjenja # postižemo sekvencom **Esc (s#B** konkretno za polucrna slova treba konstrti **Esc (s3B** je parametar na osnovu koga su laserski štampači zaradili dobar deo svoje reputacije – uz sasvim umerena ulaganja u *soft* fontove možete da štampate bez velikom tajmsom, optimom garamondom Komanda **Esc (s#T** zahte va da daji ispis bude slovima tipa # pri čemu se kodovi (0-255) formi raju u skladu sa tabelom 4 Radi se o familijama slova koja je *Hewlett Packard* ozvanicio ali mnogi drugi proizvođači fontova rezervisu dodat na kodova za svoja dizajnerska remek dela što ponekad stvara haos Kod nas je uobičajeno da YU Helvetica i YU Times nose brojeve stan dardna helvetike i tajmsa-a (4 odnosno 5) dok se za ćirilicu u helvetici i tajmsu rezervišu kodovi 252 i 254

Da bi se haos ublažio *Hewlett-Peckard* je počevši od *LaserJet IID* prošino polje *typeface* u zaglavlju fonta na dva bajta sa idejom da niži bajt oznacava familiju slova a viši bajt proizvođača lonja, na taj bi se način izbegla sva dupliranja Ova konvencija se međutim i dalja veoma retko koristi

Orijentacija *LaserJet II* je oštro razlikovao *portrait* i *landscape* orijentaciju ispisa pa su se za svaku od njih morali koristiti i odgovarajući fontovi – *portrait* lonjovima se nije moglo pisati na *landscape* stranici i obratno *LaserJet III* automatski prilagođava svaki postojeći lonj svako od orijentacija pa je ova komanda poslala besmislena Pomnjemo je ipak iz razloga kompatibilnosti **Esc &l0O** (malo L broj 0 veliko slovo O) bira *portrait* a **Esc &l1O** (malo L broj 1 veliko slovo O) *landscape* fonf

PRIORITETNI KRITERIJUMI

Ukoliko po prijemu duge kontrolne sekvence *LaserJet* pronađe fonl koji potpuno zadovoljava zahteve stvar je sasvim jednostavan – lonl će biti izabran Ukoliko međutim ni jedan od raspoloživih fontova ne zado volji sve zahteve *LaserJet* preme onome što piše u dokumentaciji *ak tivira font koji je najsličniji traženom* U nekim slučajevima vredi znati kako ovo odlučivanje teče što je najlakše razumeti na primeru

Pretpostavimo da su u RAM *LaserJet-a* učitani sledeći fontovi

Parametar	Font 1	Font 2	Font 3	Font 4
Set simbola	Roman-8	USASCII	USACII	USASCII
Proporcionalnost	prop	prop	neprop	prop
Širina slova	10	10	17	10
Visina slova	12	12	8	12
Kurziv	ne	ne	ne	da
Zacrnjenje	0	0	0	0
Familija slove	Times	Times	Helvetica	Helvetica

Štampaču je poslata komanda **Esc (0U Esc (s1P Esc (s10H Esc &l12V Esc (s0B Esc (s0S Esc (s4T** (zehvaljujući metodama kombino vanja istorodnih sekvenci o kojima smo govorili pre par meseci ova komanda bi mogle da se skrati ali ovako je jasnija) čima je zahtevan font sledećih karakteristika

Parametar	Vrednost
Set simbola	USASCII
Proporcionalnost	prop
Širina slova	10
Visina slova	12
Kurziv	ne
Zacrnjenje	0
Familija slova	Helvetica

Najpre ce otpasli font 1 pošto se zahtevani set simbola ne poklapa sa realnim Sledeći se izuzima font 3 pošto je nepropoicionalan što znači da su „u igri" ostali fontovi 2 i 4 Oba su odgovarajuće širine i visine ali je jedan kurzivni a drogi iz pogrešne familije Tvorci PCL-a su kurzivu dodelili viši prioritet pa će lonl 4 biti eliminisan a lonl 2 izabran – štampaće se lajms umesto helvetike ali su bar ostali parametri korek tni

Pokušajmo sada sa malo težim primerom sekvencom **Esc (0U Esc (s1P Esc (s14H Esc &l10V Esc (s0B Esc (s1S Esc (s6T** je zahtevan lont sledećih karaklenstika

Parametar	Vrednost
Set simbola	USASCII
Proporcionalnost	prop
Širina slova	14
Visina slova	10
Kurziv	da
Zacrnjenje	0
Familija slova	Gothic

"Rezonovanje" *LaserJet-a* teče otprilike ovako lont 1 nema odgova rajući set simodla font 3 je nepropoicionalan pa opet ostaju fontovi 2 i 4 Ni jedan od njih ne zadovoljava trazenu širinu ili visinu slova pa se po ovom kriterijumu ne moze izvršiti izbor ispluje se stil ispisa i lont broj 2 (nije kurzivan) otpada Biće izabran lonl 4 bez obzira na pogrešnu familju slova.

LaserJet III je u čitavu priču uveo još jedan faktor izvorište fonta Ukoliko, dakle, štampač pronađe više (bar dva) fontova koji podjednako zadovoljavaju uslove najpre će se opredeliti za bit mapirani *soft* lont sa najnižim identifikacionim brojem Ako takvog nema opredeljuje se za skalabilni *soft* font pa za bitmapireni lont iz kartridža (levi kartridž je prioritetan u odnosu na desni) skalabilni font iz kartridza i tek onda za fontove iz ROM a – prioritet ponovo ima bit mapirani lonl dok ce sa skalabilni font iz ROM a konstiti tek ako nema baš ničeg drugog

Ne treba imati mnogo programerskog iskustva da bi sa zaključilo kako „galanje" o lontovima tesko moze da proizvede dobre razultate – programer uvek žali da bude siguran da će neki konkretan font biti iza bran Zato je počevši od *LaserJet+* predviđeno dodeljivanje identifika cionih brojeva fontovima

Identifikacioni broj je bilo koji ceo broj između 0 i 32767 Ukoliko se radi o *soft* lontu broj mu dodeljujamo pre nego što smo ga uopste defi nisali tj poslali štampaču – ovu funkciju automatski obavljaju programi LOADFONT i AUTODL iz paketa *FontGen* Ukoliko se radi o nekom od lontova iz ROM a ili kartridža najpre ćemo ga izabrati koriscenjem se kvence slične nekoj od pomenutih (bejzik program sa slike 5 ilustruje izbor osnovnih bit mapiranih lonjova iz ROM-a *LaserJet a* III) a onda izvršiti **Esc *c#D**, gde je # broj koji dodeljujemo fontu – na ovaj način je identifikacioni broj # proglašen aktuelnim Zatim izvrsavamo **Esc *c6F** i lako aktuelnom fontu dodeljujemo aktuelni broj dovrsavajuci krei ranje takozvanog privremenog (*temporary*) identifikacionog broja – prva sledeća *reset* komanda ce anulirati dodeljivanje Ukoliko zelimo da identifikacioni broj „preživi" svaki *reset* **Esc *c6F** cemo zamenti sa **Esc *c6f5F** Svi identifikacioni brojevi i *soft* fontovi se, jasno gube kada is ključimo laserski stempač

Komanda **Esc *c#F** obezbeđuje različite operacije sa fontovima u zavisnosti od vrednosti argumenta #

Esc +c0F briše sve *soft* fontove i njima dodeljene identifikacione brojeve, kao i identifikacione brojeve svih fontova iz ROM-a. Pri bosanju se ne obraca pažnja na to da li je font (identifikacioni broj) označen kao stalan ili kao privremen

Esc +c1F briše sve privremene (*temporary*) fontove kao i privremene identifikacione brojeve dodijene ROM fontovima

Esc +c2F briše trenutno izabrani font

Esc +c3F briše poslednji karakter koji je definisan u fontu koji je trenutno izabran. Ova funkcija se koristi izuzetno retko – obično se pokazuje da je nepotreban kompletan font a ne samo neka slova u njemu

Esc +c4F proglašava izabrani font privremenim

Esc +c5F proglašava izabrani font permanentnim

Esc +c6F dodeljuje tekuci identifikacioni broj tekućem fontu. Dodeljivanje je privremeno, da bi dodeljen broj preživeo *reset* treba izvršiti **Esc +6f5F**

Brisanje fonta je u izvesnom smislu konfliktna operacija – čin brisanja označava da nam je memorija koju je zauzimao font potrebna za nešto drugo i to odmah. Ukoliko se u baterij štampaca nalazi neki teksti koji sadrži font koji se briše, taj tekst mora odmah da se štampa jer će po brisanju fonta biti kasno. Zato će *LaserJet*, ukoliko se na tekućoj strani nalazi nešto teksta ispisanog obrisanim fontom, najpre štampati stranu (ako ona nije kompletna, sami smo krivi!) i tek onda početi da prima komande racunara. Ukoliko na tekućoj strani nema nikakvog teksta pisanog fontom koji sa briše, *LaserJet* će proveriti da li se u baleru nalaze još neke strane čije je štampanje u toku, eko ih ima, prijem komandi od računara će biti zaustavljen dok se ove stranice ne ispišu. Ove diskusija objašnjava izvesnu anomaliju koju je sigurno primetio svaki korisnik *Ventura*. Štampa se nekoliko strana a onda štampica miruje dok se kompletan baler ne isprazni. To je, naime, trenutak u kome je interni menadžer fontova zaključio da neke od fontova treba ukloniti iz RAM-a. Štampaca

Pošlo je fontu dodeljen identifikacioni broj ispisivanje odgovarajućim slovima inicirarno sa **Esc** (*X gde je * identifikacioni broj potrebnog fonta

Ostalo je još da pomenemo prvi (!?) melod za izbor fontova koji se izuzetno retko koristi. Konstrukteri PCL-a su obezbedili definisanje dva fonta od kojih će se jedan zvati „primarni" a drugi „sekundarni". Pošlo su fontovi definisani sa SI aktiviramo prvi a sa SO drugi – zgodno je što fontove zamenjujemo slanjem jednog jedinog bajta, ali je nezgodno što nikada nismo sigurni koji je font trenutno aktivan (komunikacija sa laserom je jednosmerna) i što su dva fonta iskreno rečeno premalo da se bilo šta uradi. Komande SI i SO su uvršćene u PCL jedino zbog kompatibilnosti sa nekim ranijim štampačima, pa im necemo posvetiti posebnu pažnju. Iz slike 6 (mala rekapitulacija komandi o kojima smo ovoga meseca govorili) vidimo da se rad sa primarnim i sekundarnim fontom uglavnom razlikuje po tome što je otvorena zagrada zamenjena zatvorenom – **Esc (*X** na primer proglašava font * primarnim a **Esc)*X** sekundarnim

U sledecem nastavku upoznacemo komande koje se koriste za slanje fontova štampaču (*download*) i strukturu SFP datoteka koje se isporučuju uz program *Ventura Publisher*. Istovremeno ćemo napisati program koji analizira zaglavlje SFP datoteka i iz nja vadi podatke u obliku pogodnom za uključivanje u naše programe

PROCESORI 286

Nastavak se strana 13

Sve u svemu, vreme procesora 80286 sve brže prolazi. Više nikome na bismo preporučili kupovinu ovakve mašina, sem za puku obradu lakših, ili pak kao čvorište u nekoj mreži, gde će svoje zadatke sigurno obavljati sasvim lepo. No, ljudi kojima treba računar kod kuće i u profesionalne svrhe, sem ako nisu daktilografi bez ambicija ka ičemu višem od prostog unosa teksta, ove mašina, uprkos brzini, pouzdanosti i naravno ceni sve manje zadovoljavaju.

*Prosta matematika. Prema trenutnim cenama u Minhenu, računar sa procesorom SX na 20 MHz jeste nešto sporiji (10-15%) od sličnog sa procesorom 80286, ali može mnogo toga što ovaj drugi ne može nikako. Razlika u ceni na celu mašinu u bazној konfiguraciji je nešto manja od 20%, a dozvolićete, to nije mnogo da biste sebi obezbedili sigurnu budućnost. Očekujemo da će se ova razlika dosta brzo smanjiti, jer je AMD prikazao svoju verziju procesora SX na 20 i 25 MHz, konkurencija će nužno oboriti cenu.

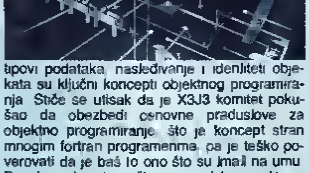
Dakle naša preporuka je – kupite SX. Jeste sporiji, ali može što ni jedna mašina sa 80286 ma kako brzim, ne može.

FORTRAN 90

Nastavak se strana 35

Razvoj softvera često zahteva da se kod ponovo piše iz početka („iz ničega"). Potprogrami ne predstavljaju dovoljno fleksibilna gradivna blokove za modernu programsku arhitekturu.

Trenutna filozofija programiranja diktira jedan objektni pristup u koma podaci imaju izvanrednu važnost. *Fortran 90* ima izvedene tipove podataka kojima manipuliše korisnik, što je poznato kao apstraktan tip podataka (*abstract date typing*). Izvedeni tipovi podataka mogu u sebi obuhvatati druge ranije definisane, izvedene tipove podataka – da li je to vrsta nasleđivanja (*inheritance*)? **Pokazivači** i **Ciljevi** uvedeni su da omoguće stvaranje privremenih veza između različitih objekata tokom izvršavanja programa – da li je to jednostavan oblik identiteta objekata (*object identity*)? Apstraktni



tipovi podataka, nasleđivanje i identiteti objekata su ključni koncepti objektnog programiranja. Stiče se utisak da je X3J3 komitet pokušao da obezbedi osnovne preduslove za objektno programiranje, što je koncept stran mnogim fortran programerima, pa je teško poverovati da je baš to ono što su imali na umu. Posebno zbog toga što se u celokupnoj literaturi trenutno na raspolaganju o *Fortranu 90*, čiji su autori članovi komiteta odgovorni za novi standard, ni jednom rečju na spominje objektno programiranje. Možda članovi komitela nisu ni svesni (!?) šta su stvorili, možda je sve to slučajna greška.

Uprkos mirisu kompromisa, X3J3 komitet svakako se potrudio da od fortrana načini programski jezik dostojan 90-tih. Da li će u tome uspeti ili ne, zavisi od toga koliko je još fortran programera ostalo koji su zainteresovani za razvoj ovog jezika. Vreme utrošeno za definisanje novog standarda, kao i uključivanje nataknutog *Fortrana 77* u njega, verovatno će navesti mnoge programere da ignorišu revolucionarne karakteristike *Fortrana 90*. Prvi kompajleri za novi fortran bi trebalo da su se već uveliko pojavili, jer su najavljeni za jun '91. Sveteko će biti interesantno poredili vreme potrebno prvom komercijalno raspoloživom fortran kompajleru da se pojavi, sa svojevremenom žurbom iskazanom od strane proizvođača C++ kompajlera. I eto, kada se zabava završila, i kada su svi dobili priliku da prespavaju nad uzbuđenjima u zadnjih 5000 dana, valjda su zaključili da nisu uradili loš posao. Rezultat toga je moćan, fleksibilan i strukturan, moderan jezik, pogodan za nove tipove hardvera, posebno paralelne mašine. Na kraju krajeva, bilo koji jezik sa naredbom ELSEWHERE ne može biti baš sasvim loš!

Priredio. Kragi Aleksandar

MEKINTOŠ 7.0

Nastavak se strane 47

Vraćata se zatim u prvi program te nastavljate sa radom na slici. Od sada će se svaka promjana slike u prvom programu reflektirati na sadržaj slika u drugom. Ovo je posebno korisno pri radu u mreži kada se više korisnika može pretplatiti na neki dokument te aktivno manipulirati sa tim dokumentom dok istovremeno njegov izdavač radi na njegovom dovršavanju. Od ostalih novileta koje je Apple ugradio u novi sistem svakako bi izdvojio i TrueType koji sada omogućava rad sa konturnim fontovima čiji je prikaz jednako oštar na ekranu kao i na bilo kojoj ispisnoj jedinici, bez potrebe za PostScriptom i ATM-om.

Ono što bih posebno naglasio, te sam iz tog razloga i oslavio za kraj, možda će zvučati smiješno ali je izuzetno korisno i funkcionalno. U novi je sistem ugrađena tzv *balonska pomoć*. Radi se naime o tome da će vas sistem, aktiviranjem ikone u gornjoj, desnoj strani desktopa, opslužiti informacijama u vidu balona a koje možete videti u bilo kojem stupu. Dovoljno je samo mišem odabrati neki element na ekranu i kraj njega će se otvoriti balon sa odgovarajućim, popratnim tekstom (pogledajte sliku). Važno je napomenuti da će i buduća aplikacije sadržavati pomoć u vidu balona. Zvuči i izgleda smiješno ali pomoću ovih balona konšlenje Macintosha bi će teško kao i čitanje stripa o Miki Mausu.

Sistem 7.0 se ne odlikuje revolucionarnim poboljšanjima i izvanzemaljskim funkcijama. Gotovo svi novougrađeni atati već su negdje viđeni ili se odavno koriste na drugim sistemskim platformama. Pa ipak, Apple se potrudio da odabere najkorisnije, te da to integrira u kompaktnu cjelinu stvorivši na taj način izuzetno snažnu radnu okolinu. Njegov se sistem sada odlikuje izuzetnom funkcionalnošću, ali i naglašenom jednostavnošću, faktorima koji će oduševiti stare Macintosh korsnike, ali zasgurno privući i veliki broj novih.

KORISNA ADRESA

Apple Centar NOVEL
41000 Zagreb, Mirana Rečetara 44
Tel. 041/ 34 75 14, Fax. 041/ 34 79 43

SPECIJALNI KOMERCIJALNO-INFORMATIVNI BLOK

akcije • marketing • publicitet

U BLOKU „AKCIJE — MARKETING — PUBLICITET" OVOG PUTA PREDSTAVLJAJU VAM SE FIRME:

KODEKS, Zagreb	79	INTERSOFT, Beograd	96
GENELEC, Tuzla	80	MIKROSAM, Skopje	96
GAMA ELECTRONICS, Beograd	81	GOSTOL-GOAP, Nova Gorica	97
COMTEC, Novi Sad	82	TRIAS, Ljubljana	98
ELSI SISTEMI, Čakovec	83	DEWETRON, Radeče	98
COMPUTERWORLD-ACS, Darmstadt/Bgd	84	IBC, Trst	99
ŠUTLIĆ, Beograd	85	MOST, Beograd	100
BSV, Prijepolje	86	MIKRONOVA, Novi Sad	101
MZ, Beograd	86	KMA, Beograd	102/103
MICRO ENGINEERING, Beograd	87	MEDIS, Ljubljana	104
HSM, Zagreb	87	COMTRAD, Düsseldorf/Bgd	105
CENTAR ZA PERM. OBR, Beograd	88	OSA, Beograd	106/107
COMPUTER HIT, Ilidža	89	ABV GRAPHICS, Beograd	108
BEST, Beograd	90	MOST, Beograd	109
MICROLINE, Zagreb	94	REPRO, Ljubljana	110
MLACOM, Ljubljana	92/93	METRONIK, Ljubljana	111
PERIHARD, Zagreb	94	ENERGODATA, Beograd	112
INFOTRADE, Koper	95	PERPETUUM, Zagreb	113
MRAK, Ljubljana	95	MCH, Maribor	114/115
ELHARD, Kula	95	ATLANTIS, Ljubljana	116
YUCO, Novi Sad	96		

Povodom sajma računarske opreme u Zagrebu 22—26. 10. 1991. redakcija „Računara" po tradiciji priprema specijalni dodatak

Sajamski vodič

INTERBIRO — INFORMATIKA 1991.

Dodatak će biti štampan kao sastavni deo časopisa, ali posebno formiran, sa vlastitom naslovnom stranom, sadržajem i pregledom svih izlagača po paviljonima i štandovima. Biće štampan na povećanom broju strana, u najkvalitetnijoj tehnici (tabačni off-set) i na najfinijem premaznom papiru. „Računari" broj 76 sa specijalnim dodatkom pojaviće se u prodaji sredinom oktobra.

Svoje mesto u ovom vodiču će naći mnoga preduzeća koja izlažu na ovoj uglednoj manifestaciji, kao i mnoge firme koje ne učestvuju na sajmu ali žele da iskoriste ovu izuzetnu priliku i pojave se u specijalnom dodatku koji uvek privlači veliku pažnju javnosti — s obzirom da predstavlja najveću računarsku berzu i najkompletniji računarski marketinški katalog sa cenama.

CENOVNIK OGLASNOG PROSTORA U SAJAMSKOM VODIČU
(isti kao u redovnim brojevima „Računara")

- poslednja strana korica **46.000 O,** • druga ili treća strana korica **44.000 O,**
- 1/1 unutrašnja kolor strana **36.000 O,** • 1/2 unutrašnje kolor strane **25.000 D,**
- 1/1 crno-bela strana • **30.000 O,** 1/2 crno-bele strane **20.000 D,**
- 1/4 crno-bele strane • **15.000 O** 1/8, crno-bele strane **10.000 D**

Za sve komercijalne aranžmane u Sajamskom vodiču INTERBIRO 91. izvolite nam se javiti na

tel. 011/653-748, fax 011/648-140, modem 011/648-899
(osoba za kontakt: Esad Jakupović)

Rok za rezervaciju prostora je *20. septembar*, a za dostavu materijala *25. septembar*.
Javite nam se na vreme!

Samo najbolje!



EVEREX
EVER for EXcellence

Vrhunski američki proizvodač PC sistema
i opreme. Najsposobniji sistemi bez ograničenja!
EVEREX STEP, TEMPO, MEGACUBE.
Serije 286, 386, 486.
Komunikaciona i grafička oprema.

FACIT

Visoko kvalitetan švedski proizvodač,
pouzdane i izdržljive računarske opreme.
Iglični štampači, laserski štampači, terminali,
NC periferije, čitači i bušači trake.
Laserski štampači već od 49.900 din.





CONNER

Najkvalitetniji američki proizvođač tvrdih diskova.
Vrhunska tehnologija, izuzetni kapaciteti, male
dimenzije. Izdržljivost i otpornost na udarce.

Multi Project d.d.
CELOVITI PROJEKTI S PODROČJA INFORMATIKE IN RAČUNALNIŠTVA

Tržaška 132, 61000 Ljubljana
Tel.: 061/274-361 int.: 715
061/268-341
Fax: 061/274-568

KODEKS
PODUZEĆE ZA INFORMATIČKU DJELATNOST

Medulićeva 1, 41000 Zagreb
Tel/Fax: 041/421-194
Tel.; 041/428-961
041/428-973



GENELEC

ELEKTRONIKA & INFORMATIKA

75000 TUZLA
Zlokovac 3B
tel. 075/230-878, lokal 126, 231-557
fax. 075/237-825, 237-778

11000 BEOGRAD
Jurija Gagarina 115
tel. 011/179-967
fax. 011/179-967

PROMPTNA ISPORUKA RAČUNARA

PONUDA RAČUNARA

AT 12 MHz/bez DISKA — 25 850 DIN

slim line, 286/12 MHz, 512 KB,
FDD 1.2 MB, 2 × ser. + 2 × paralel,
c/b monitor, YU znaci

AT 12 MHz/45 MB — 39 200 DIN

slim line, 286/12 MHz, 1 MB, 45 MB
HDD (1:1), FDD1.2 MB, 2 × ser. +
2 × paralel, c/b monitor, YU znaci

AT 16 MHz/45 MB — 39.830 DIN

slim line, 286/16 MHz, 1 MB, 45 MB
HDD (1:1), FDD 1.2 MB, 2 × ser. +
2 × paralel, c/b monitor, YU znaci

386 SX 16 MHz/45 MB — 53 990 DIN

slim line, 386 SX/16 MHz, 1 MB,
45 MB HDD (1:1), FDD 1.2 MB, 2 × ser
+ 2 × paralel, c/b monitor, YU znaci

386 25 MHz/85 MB — 79 100 DIN

Mini Tower, 386/25 MHz, 2 MB,
85 MB 19 ms HDD AT BUS, FDD 1.2 MB, 2 × ser
+ 2 × paralel, c/b monitor, YU znaci

PONUDA ŠTAMPAČA DATAPRODUCTS

MATRIČNI		
9030	(A_4, 9 pin, 300 cps, YU, paral.)	28.850
9040M	(A_3, 9 pin, 300 cps, YU, paral.)	34.100
9044M	(A_3, 24 pin, 300 cps, YU, paral. + ser.)	42.990
Star LC 20	(A_4, 9 pin, 180 cps, YU set, kabl)	15.870
Star LC 15	(A_3, 9 pin, 180 cps, YU set, kabl)	20.060
Star LC 24-10	(A_4, 24 pin, 170 cps, YU set, kabl)	24.480
Star FR 15	(A_3, 9 pin, 300 cps, YU set, kabl)	34.110
LASERSKI		
LZR-650	(6 str/min, 512 K)	63.260
Xerox 4045DP	(10 str/min, opcija kopir uređaja)	84.900
LZR-1240 E	(12 str/min, 1 MB)	nazvati
LZR-1260 E i Postscript	(12 str/min, 4 MB)	nazvati
LZR-2655/H P2000	(26 str/min, A_3 format)	nazvati
LZR-2665 i Postscript	(26 str/min, A_3 format)	nazvati
LINIJSKI		
LB 325	(300 lpm)	nazvati
LB 625	(700 lpm)	nazvati
LB 1015	(1000 lpm)	nazvati
LB 1515	(1500 lpm)	nazvati
LM 300 Table Top	(300 lpm, matrix)	nazvati
LM 615	(600 lpm, matrix)	nazvati
LM 915	(900 lpm, matrix)	nazvati
LM 1200	(1200 lpm, matrix)	nazvati

U CIJENE NIJE URAČUNAT POREZ NA PROMET

PONUDA DIJELOVA

KUĆIŠTA	DIN
Baby 8 PS 200 W	4 730
Mini Tower 8 PS 200 W	5 680
Midi Tower 8 PS 200 W	5 580
386 Tower & PS 220 W	7 890
486 Tower 8 PS 250 W (sa 2 ventilatora)	7.560 10.580
MATIČNE PLOČE	
AT 286/12 MHz	5.160
AT 286/16 MHz	6 710
386 SX/20 MHz	11 660
386/20 MHz	19 950
386/25 MHz	26 670
386/33 MHz Cache 64 KB	35.080
486/25 MHz AT BUS	76 100
I/O KARTICE	
AT I/O (2×ser.+paralel)	901
AT I/O & adapter	1 890
Paralel port card	474

KONTROLERI (sa kablovima)	DIN
AT BUS (IDE) Adapter 8 FDC	1 170
ESDI WD 1007V-SE2 (1:1)	10 660
SCSI WD 7000 FASST	13 220
STRIMERI WANGTEK	
3040 FK 40 MB, Int. 3.5"	17 070
5099 EK 80 MB Int. 5.25	30 760
5150 PK 150 MB, Int. 5.25"	38 480
DISKETNI POGONI	
5.25 1.2 MB	3 420
3.5 1.44 MB	3 080
TVRDI DISKOVI	
ST 157A 1 AT BUS 45 MB/28 ms	11.630
Maxtor 8051A 42 MB/28 ms	12 350
ST 1102A AT BUS 69 MB/19 ms	19 840
Maxtor 7080A 61 MB/19 ms	21 380
ST 1144A AT BUS 125 MB/16 ms	26.660

TVRDI DISKOVI (nastavak)	DIN
Maxtor ND 213A AT BUS 213 MB/15 ms	51.300
Maxtor 4170 ESDI/SCSI 158 MB/16 ms	72.080
Maxtor 4380 ESDI/SCSI 336 MB/16 ms	57.600
Micropolis 1554-7 ESDI 161 MB/16 ms	76.960
Micropolis 1664-7 ESDI 340 MB/14 ms	76.840
Micropolis 1674-7 SCSI 161 MB/16 ms	52.160
Micropolis 1664-7 SCSI 340 MB/14 ms	76.960
TASTATURE	
XT/AT, YU/ASCII, 102 tastera, klik	2 420
MEMORIJE	
DRAM 4164-10	90
DRAM 41256-10	325
DRAM 41464-10	120
DRAM 44256-10	90
DRAM 411000-07	285
SIMM MODUL 256*9-08	600
SIP MODUL 256*9-08	1.060
SIMM/SIP MODUL 1 M*9-07	3.250

KOPROCESORI	DIN
80387-10 MHz AMD	4 640
80287 XL	11 940
80387SX-16	17 070
80387-25	24 590
80387-33	30 448
GRAFIČKE KARTICE	
MGP	1 170
VGA/16 800×600	5 150
Super VGA 1024×768	7 400
MONITORI	
14" Mono, amber/cb	5 370
14 VGA Mono 1024×768	6 716
14" VGA Kolor 1024×768	19.950
OSTALO	
Miš Dulck QM-E1	1 840
Modem Discovery 2400	6 190

ZASTUPNICI

„C-SOFT"	LESKOVAC,	tel. 016/47-067,	fax. 016/47-830	„ARDI ENGINEERING"	PRIŠTINA,	tel. 038/22-981,	fax. 038/23-578
„BEE LINE"	NIŠ,	tel. 018/323-173,	fax. 018/47-424	„PIM"	RIJEKA	tel. 051/429-361,	fax. 051/429-361
„SPASCHEX"	NOVI SAD	tel. 021/341-685,	fax. 021/341-685	„LAKOL"	ŠIBENIK,	tel. 059/33-589	fax. 059/28-781

GAMA Electronics Beograd
Tel: 011/332 275, Fax: 011/335 902
Radno vreme: pon.-petak 9-17
GAMA Electronics Trade Handels GmbH
Tel: 99/49/89/577-209
Fax: 99/49/89/570-4379



GAMA KOMPJUTERI - STANDARD KOJI PREVAZILAZI VAŠA OČEKIVANJA

Ugled koji GAMA na našem tržištu ima, direktno proističe iz naše neprestane težnje da vam ponudimo kompjuter u koji se bezrezervno možete pouzdati.

Da smo na dobrom putu svedoče:
garancija od 24 meseca
i
naše dugogodišnje prisustvo i stalan rast na jugoslovenskom i nemačkom tržištu kompjutera.

GAMA KOMPJUTERI BRZI SU I SNAŽNI KOLIKO TO VAMA TREBA, A POUZDANI UVEK I PRE SVEGA

ComTec d.o.o.

Preduzeće za informatički inženjering i automatizaciju
Novi Sad, Vojvođanskih brigada 7/IX
Telefon/Fax: 021/22.258

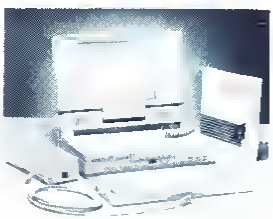
LAPTOP - NOTEBOOK



VERIDATA EL 386
samo 2kg!

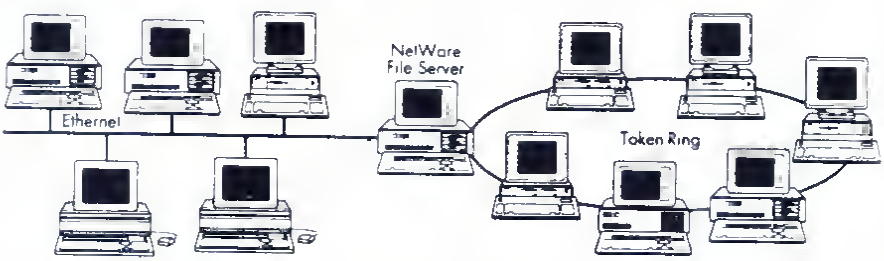


FLYER 286 FH
FLYER 386 FH



VERIDATA LP 286
VERIDATA LP 386

NOVELL



ComTec d.o.o. je autorizovani prodavac *NOVELL* proizvoda
Nudimo Vam i standardne AT 286 /386/486 računare i š irok izbor aplikativnog softvera prilagođenog radu u mreži

Dileri, javite se! Povoljan rabat!

sistem

ELSI

Specialix

INTELIGENTNI ADAPTERI ZA SERIJSKU KOMUNIKACIJU

SI/4	adapter s 4 serijska kanala	1.540 BODOVA
SI/8	adapter s 8 serijskih kanala	2.160 BODOVA
SI/16	adapter s 16 serijskih kanala	3.090 BODOVA
TA/4	4 dodatna serijska kanala	925 BODOVA
TA/8	8 dodatnih serijskih kanala	1.235 BODOVA
OSTALI SPECIALIX PROIZVODI		CIJENA NA UPIT

SCO
THE SANTA CRUZ OPERATION

UNIX/XENIX PROGRAMSKA OPREMA

SCO UNIX System V/386 Release 3.2v2	Operating System Development System	2.135 BODOVA 2.490 BODOVA
SCO XENIX System V 386	Operating System Development System	1.928 BODOVA 1.990 BODOVA
SCO TCP/IP (UNIX)	Runtime System	995 BODOVA
SCO NFS (UNIX)	Unlimited System	1.280 BODOVA
Microsoft Word (UNIX/XENIX)		2.135 BODOVA
OSTALI SCO PROIZVODI		CIJENA NA UPIT

James River Group

PROGRAMSKA OPREMA ZA UNIX-DOS POVEZIVANJE

ICE.TEN	Povezivanje DOS-UNIX do 8 PC računala (serijska veza)	560 BODOVA
ICE.TEN.PLUS	Povezivanje DOS-UNIX za više od 8 PC računala (serijska veza)	750 BODOVA
ICE.TCP	Povezivanje DOS-UNIX (Ethernet-TCP/IP)	378 BODOVA

Locus
Computing Corporation

PROGRAMSKA OPREMA ZA UNIX-DOS UMREŽAVANJE

PC-Interface DOS Bridge	430 BODOVA
PC-Inteeface DOS Rridge s podrškom MS Windows 3.0	515 BODOVA
PC-Interface UNIX Server	578 BODOVA
PC-Xsight X Window Servee za DOS	945 BODOVA
PC-Xsight X Window Server s TCP/IP za DOS	1.090 BODOVA

SVE CIJENE IZRAŽENE SU U BODOVIMA. VRIJEDNOST JEDNOG BODA IZNOSI. 1 BOD = 1 DEM

PLAĆANJE SE VRŠI U DINARSKOJ PROTUVRIJEDNOSTI DEM NA DAN PLAĆANJA PO TEČAJU BURZE U ZAGREBU

(05.09.1991. = 23 DINARA)

ČAKOVEC, I. MAŽURANIĆA 2
TEL. (042)816-444 FAX. 812-370

trade

ELSI

ZAGREB, J. GOTOVCA 1
TEL. (041)456-155 FAX. 410-448

computerworld

Hindenburg str. 56, 6100 Dermstadt
TEL: 9949/6151/31 56 68, 31 56 36
FAX: 9949/6151/31 45 00, 31 34 02

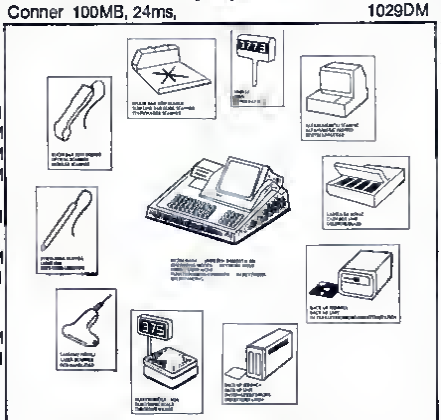
Novo otvoreno predstavništvo-ured: 41000 Zagreb, Rediceva 11, tel:041/421-873, 435-448
Novo otvoreni izlozbeni i prodajni prostor; 41000 Zagreb, Margaretska 2/1, tel:041/423-604

POZIVAMO PODUZECA I SERVISERE ZAINTERESIRANE ZA RAD SA NAMA DA NAM SE JAVE

NASA AKTUELNA PONUDA

MOTHERBOARD	
80286/12Mhz,G-2	175DM
80286/16Mhz, G-2	239DM
80386 /16Mhz,SX	609DM
80386/25 w/o Cache	1089DM
80386/25,64Cache	1409DM
80386/33,64Cache	1559DM
60486/25,128Cache	3095DM
GRAPHIC CARDS	
monochrom+LPT1	34DM
Vga 8bit/256KB	114DM
Vga 16bit/256KB	169DM
Vga 16bit/512KB	199DM
CONTROLLER CARDS	
MFM,1:1,2Hdd/2Fdd	89DM
AT-BUS, 2Hdd/2Fdd	34DM
WD 1006MM2	179DM
WD 1006SR2	169DM
WD 1007ESDI	359DM
MULTI I/O CARD	
2ser+1par+1game	25DM
At-bus adap+multi I/O	65DM
XT/Multi I/O card	65DM
FLOPPY DRIVES	
Mitsumi/YE-DATA	
1,2MB ili 1,44MB	119DM
TEAC 1,44MB	134DM
TEAC 1,2MB	129DM
KEYBOARDS	
YU-sa ili bez Click-a	72DM
CHERRY US	129DM
Chicony US	72DM
KUCISTA	
DTP,200W bez LED	135DM
DTP,200W sa LED	199DM
Mini-tower,200W	239DM
TOWER case, 220W	379DM
STREAMER	
40MB Wangtek,int.	595DM
40MB Wangtek,ext	769DM
COPROCESORS	
80287/XL/Intel	389DM
80387/SX/Intel	559DM

HARD DRIVES	
Maxtor 8051A 42MB, 28ms 2 god.garancije	409DM
Seagate ST 157A,42MB, 28ms,1god garancije	395DM
NEC 3142,42MB,28ms, 1god.gar	475DM
Conner 100MB, 24ms,	1029DM
Maxtor 7080A,82MB, 19ms 2 god.gar	695DM
Seagate ST1102, 82MB, 28ms 1god.gar	679DM
NEC D5665,179MB, 16ms 1god.	1299DM
Maxtor MXT 4380S/E, 382MB,16ms	2649DM
Isporucujemo HDD's CONNER, Quantum	



PERIFERIJE ELKAS-a REGISTAR BLAGAJNE • PRINT-OF-SALE NETWORKING • REGISTER KASE PETZWERK

POINT OF SALE ELKAS 90 ELEKTRONSKA KASA BAZIRANA NA PC SISTEMU

80286/16Mhz Intel Processor, ugraden POS-BON printer EPSON TM 262 sa interface card. 1MB RAM 1*1,44 FDD, Maxtor, 42MB HDD 256KB 8bit VGA, orig. IBM monitor 9", 3 slobodna slota za prosirenje network compatible ladica za novac, 200W PS **3798 00DM**

LCD DISPLAY 2*16 znakova + interface/OPTREX	195,00 DM
BARCODE LIGHT PEN + DECODER / DATA LOGIC	245,00DM
BARCODE TOUCH SCANNER /DATA LOGIC	1235,00DM
EPSON TM 930 MULTIFUNKCIONSKI PRINTER SA BON-JOURNALOM-SLIP PRINTEROM	1565,00DM
POWER SUPLY ZA EPSON TM 930	259,00DM

Pored toga nudimo rjesenja za mobilnu obradu BAR-CODA, te obradu u okviru svih postojecih mreza.

MONITORS	
IBM Vga monoc 9"	495DM
Monoc 14" III amber	195DM
Monoc VGA 14"	249DM
Mon.VGA, MG-11,14"	419DM
Vga Col 800*600, 14"	609DM
Vga Col 1024*768, 14"	649DM
Vga Col 1024*768, 19"	2179DM
TERMINALI	
Wyse 60	1009DM
Ourne OVT 101+	889DM
Ourne PCT EV	969DM
NETWORK CARDS	
Ethernet NE1000or	389DM
Ethernet NE2000or	459DM
Ethernet NE1000com	219DM
Ethernet NE2000com.	279DM
MEMORY	
41256/80	3,20DM
511000/60	1,20DM
514256/80	2,90DM
256*9/70 SIMM	34DM
256*9/70 SIPP	36DM
1MB*9/70SIMM	109DM
1MB*9/70SIPP	111DM
MATRIX/LASER PRINTER	
Epson LX400	419DM
Epson LX850	579DM
Epson LQ400	619DM
Fujitsu DL1100	869DM
Laser Ourne WP+	2089DM
Ourne SerlesII,1,5MB	3289DM
HP Laser Jet IIP	2499DM
HP Laser Jet III	3599DM
HP DESKJET 500	1149DM
SCANNERS/MAUSES	
Genius Gmf 302	85DM
Genius Gmf 320	59DM
Logitech 256 Scanner	569DM
Chinon A-4	1029DM
HP Deskjet+	2979DM
Genius GS 4500	289DM
FAX-CARD	389DM

Garantni rok za ponudenu opremu je 12mjeseci. Rok isporuke: sa ekiedista, za dealere po dogovoru. Za ponude, eventuelne potreskoce obratite nam se direktno ili pek nasim ovlastenim serviserima.

INTERELEKTRONIK STEVANA SUPLJIKCA 45 26000 PANCEVO TEL: 013/311-150, FAX 521-088	INFORMATICKI INZINJERING, e POLJAK ZELIMIR GUNDULICEVA 61F 54 000 OSIJEK TEL: 054/41666-278, FAX/TEL: 125-371	ACS gosp. VOJISLAV PAVLOVIC HO-SI-MINOVA 27 SAM KONZALTING CENTAR 11000 BEOGRAD TEL: 011/132 219

PC AT 286, 386-SX, 386, 486
ISPORUKA ODMAH

MODEL	SC 2016	SC 2025III	SC 3020 SX	SC 3033 CACHE 64 Kb	SC 4033
PROCESOR	INTEL 266	INTEL 266	INT 386 SX	INTEL 386	INTEL 486
CHIP SET	HEADLAND	HEADLAND	HEADLAND	C&T	OPTI
RADNI TAKT	16 MHz	25 MHzIII	20 MHz	33 MHz	33 MHz
LAND TEST	21 MHz	33 MHzIII	26 MHz	58 MHz	150 MHz
KUĆIŠTE	BABY AT	BABY AT	BABY AT	BABY AT	BABY AT
NAPAJANJE	200 W	200 W	200 W	200 W	200 W
RAM	1 Mb	1 Mb	2 Mb	4 Mb	4Mb
MAX RAM	5 Mb	6 Mb	6 Mb	16 Mb	16 Mb
FLOPPY	1 2 Mb	1 2 Mb	1 2 Mb	1 2 Mb	1 2 Mb
HARD DISK	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb
HD FD CONT	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1
CORETEST	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S
SER INT	2 RS 232	2 RS 232	2 RS 232	2 RS 232	2 RS 232
PAR INT	2 CENR	2 CENTR	2 CENTR	2 CENTR	2 CENTR
TASTATURA	101 ALPS	101 ALPS	101 ALPS	101 ALPS	101 ALPS
GRAF KART	HERCULES	HERCULES	HERCULES	HERCULES	HERCULES
MONITOR 14"	AMBER ili CRNO BELI	AMBER ili CRNO BELI	AMBER ili CRNO BELI	AMBER ili CRNO BELI	AMBER ili CRNO-BELI
CENA DIN.	**38000,00**	**42000,00**	**52000,00**	**78000,00**	**108000,00**

16" COLOR MONITOR EIZO 9070S (1200*800)		*
I VGA CARD 1 MB 1024*768	50000.00 DIN	*
14 " VGA COLOR MONITOR I VGA CARD 512 Kb 1024*768	22000.00 DIN	* (ZVEZDICOM SU OZNAČENE DOPLATE U ODNOSU NA OSNOVNE KONFIGURACIJE)
HD MAXTOR 7040A 17 ms	1000.00 DIN	*
HD 65 Mb MAXTOR 7080 A 17 ms	11000.00 DIN	*
HD 130 Mb ST 1144 A IDE 16 ms	15000.00 DIN	*
HD 210 Mb CDC IDE 15 ms	30000.00 DIN	*
1 Mb RAM	3000.00 DIN	*
IIT 287—10 MHz KOPROCESOR	6500.00 DIN	(100—300% BRŽI OD INTELA 287)
IIT 287—20 MHz KOPROCESOR	8000.00 DIN	(100—300% BRŽI OD INTELA 287)
FLOPPY DISK 1.44 Mb TEAC	4500.00 DIN	
MIŠ GM 6000 (RS 232)	2000.00 DIN	
EPSON LX 400 (A4 9 PINA)	12500.00 DIN	(PRINTERI SU SA UGRADENIM YU ZNAKOVIMA ŠŽĆČĐ I KABLOM ZA VEZU SA RAČUNAROM)
EPSON LQ 550 (A4 24 PINA)	26000.00 DIN	
EPSON LQ 1010 (A3 24 PINA)	30000.00 DIN	
EPSON LQ 1050 (A3 24 PINA)	40000.00 DIN	
EPSON FX 1050 (A3 9 PINA)	30000.00 DIN	
EPSON LX 1050 (A3 9 PINA)	25000.00 DIN	

ZADRŽAVAMO PRAVO KOREKCIJE CENA BEZ NAJAVE!!!
NAČIN PLAĆANJA: 100% AVANS — ISPORUKA ODMAH.

PREDUZEĆE „ŠUTLIĆ“ & „MIKRO DIZAJN“

KOSOVSKA 32/1. SPRAT, 11000 BEOGRAD,
RADNO VREME: 9—17 h, SUBOTOM 10—14 h

TEL. 011/332-607

GARANCIJA 12 MESECI, SERVIS I ODRŽAVANJE
BEOGRAD 05. 08. '91.

CENE OPREME VAŽE ZA PRAVNA I FIZIČKA LICA.
(OD 01. 02. '91. PLAĆA SE POREZ 3%)

PROFESIONAL
COMPUTER
SYSTEMS

WEST

„INTELCOMP"
Budva
Tel 086/52 217
Fax 086/52-103

AT 286-12/44

Mainboard 80286 CPU 12MHz
Baby kućište sa 200W PS
DRAM 1 MB
Hercules karta sa 1 × PAR port
Serial port
2 × HDC 2 × FDC AT BUS (IDE)
HDD ST157A 44MB 28ms
FDD 5 25 (1 2MB)
Tastatura 102 tastera USA
Monohrom monitor

CENA 1 630 DEM

Doplata za AT286 16MHz 82 DEM

AT386SX-16/44

Mainboard 80386SX CPU 16MHz
Baby kućiste sa 200W PS
DRAM 1MB
Hercules karta r 1 × PAR port
Serial port
2 × HDC 2 × FDC AT BUS (IDE)
HDD ST157A 44MB 28ms
FDD 5 25" (1,2MB)
Tastatura 102 tastelera USA
Monohrom monitor

CENA 2 039 DEM

AT386 25/44

Mainboard 80386 CPU 25MHz
Baby kuciste sa 200W PS
DRAM 1MB
Hercules karta r 1 × PAR port
Serial port
2 × HDC 2 × FDC AT-BUS (IDE)
HDD ST157A 44MB 28ms
FDD 5 25" (3,5MB)
Tastatura 102 tastera USA
Monohrom monitor

CENA. 2.724 DEM

Za sve konfiguracije doplata za VGA
Color monitor 1024 768 i
VGA karta 1024 768 16bita 512k

CENA 752 DEM

OSNOVNE PLOČE

80286 12MHz do 4MB	205
80286 16MHz do 4MB	283
80386SX 16MHz do 8MB	614
80386 25MHz do 6MB	1 239
80386 33MHz do 8MB	1 620
80486 25MHz do 8MB	3 675
80486 25MHz cache 128K do 8MB	4 125

DRAM

41256-80	4
44256-08	18
511000-08	16
SIMM/SIPP modul 256K-80	46
SIMM/SIPP modul 1MB-80	139

KOPRDCESORI

82S87 (za AT do 20MHz)	269
83S87 16 (za 386SX do 16MHz)	419
83D87-25 (za 386DX do 25MHz)	639
83D87-33 (za 386DX do 33MHz)	666

VIDEO KARTE

HERCULES (720×348)+PAR	35
VGA (800×600)	199
VGA 16BIT 512K (1024×768)	278

FLOPI DISKOVI

1 2MB 5 25"	169
1 44MB 3 5"	168
okvir 5 25" za 3 5" FDD	45

HARD DISKOVI

SEAGATE ST157A 44MB 28ms	486
SEAGATE ST1102A, 89MB 16ms	872
QUANTUM 105MB 17ms	1 165
QUANTUM 170MB 17ms	1 868
QUANTUM 210MB 17ms	2 010

MONITORI

14"MONOHROM	240
14" VGA MONOHROM 1024×768	353
14" VGA COLOR 1024×768	752
14" VGA MULTISYNC 3D 1024×768	1.875

TASTATURA

102 TASTERA (USA)	83
102 TASTERA USA (klik)	104

ŠTAMPAČI

EPSON LX400 9PIN A4	570
STAR LC20 9PIN A4	580
STAR LC24 15 (A3, 24 pina) 200cps	1 335
SANYO PR241 (A3 24pina)	1 199
HP LASERJET IIP	3.339
HP LASERJET III	4 499

MODEMI

2400 baud INTERNI	248
2400 baud EXTERNI	293

MIŠEVI

CHIC 290-1450 dpi	60
GENIUS GM6+ (software)	83
GENIUS GS4500 (ručn skener)	390
TABLET GENIUS 12128 12×12	735

STRIMERI

COLORADO DJ10 120MB interni	1.012
COLORADO DJ20 250MB interni	1.223

KUĆIŠTA

AT baby 200W display	180
Mini tower 200W display	225
Tower 230W display	368

INTERNE KARTE

AT-BUS (IDE)	47
AT-BUS (IDE) 2×HDC 2×FDC SER-PAR	85
SERIAL KARTA	29
2×SEŠ 2×PAR 1×GAME KARTA	48

SVE CENE SU U DEM DINARSKA CENA NA UPIT
CARINA I POREZ URAČUNATI
PLAĆANJE UNAPRED ISPORUKA ODMAH
GARANCIJA GODINU DANA
OBEZBEDEN SERVIS NA CELOJ TERITORIJI SFRJ
ZA STALNE KUPCE I PORUDŽBINE PREKO 10 000 OEM RABAT 5%
VESELIĆE NAS VAŠA PORUDŽBINA

Proizvodnja „MZ" 11000 Beograd
Dubljanska br 70 Tel 011/434-612.
Fax 011 450 471

ŠTAMPAČI

EPSON

LX-850, 9-igličn A4 format 200 znakova/s
FX-850m 9 igličn A4 format 300 znakova/s
LX 1050 9-igličn A3 format 200 znakova/s
FX 1050 9 igličn A3 format 300 znakova/s
LQ-550 24 igličn A4 format 180 znakova/s
LQ-850+ 24 igličn A4 format 300 znakova/s
LQ-1010 24 igličn A3 format 180 znakova/s
LQ-1050+ 24 igličn A3 format 300 znakova/s
LQ-860 24-igličn A4 format kolor 300 znakova/s
LQ-1060 24-igličn A3 format kolor 300 znakova/s
LQ-2550 24 igličn A3 format kolor 400 znakova/s
DFX-5000 „heavy duty" 9 igličn 533 znakova/s A3 format
DFX-8000 „heavy duty" 9-igličn A3 format 1066 znakova/s
EPL 7100 laserski štampač MEGA II 2 Mb 6 str/min
EPL 7500 laserski štampač se onginalnim Adobe PostScriptom
GT-4000 color scanner
GT-6000 color scanner
GT-1000 handy scanner

Roland
DIGITAL GROUP

DXY 1100	*A3 PLOTER*
DXY 1200	*A3 PLOTER*
DXY 1300	*A3 PLOTER*
DPX-2500	*A2 PLOTER*
DPX-3500	*A1 PLOTER*
GRX 300	*A1 „ROLL" PLOTER*
GRX-400	*A0 „ROLL" PLOTER*
DPX-4600	*A0 PLOTER*
LTX 100	*A3 TERMIČKI PLOTER*
LTX 120	*A3 TERMIČKI PLOTER*
LTX 321	*A1 TERMIČKI PLOTER*
LTX-420	*A0 TERMIČKI PLOTER*
CAMM 1	*PNC 1300 PLOTER REZAČ*
CAMM 1	*PNC 1600 PLOTER REZAČ A0 FORMAT*
STIKA	*REZAČ SA UGRADENIM SKENEROM*
SKETCHMATE A4 PLOTER	
ROLAND PERA ZA PLOTERE	

EIZO

EIZO 4050	*14 MONOHROMATSKI MONITOR*
EIZO 6500	*21" MONOHROMATSKI MONITOR*
EIZO 9060S	*14 KOLOR MONITOR*
EIZO 9070S	*16 KOLOR MONITOR*
EIZO T560	*17" KOLOR TRINITRON MONITOR*
EIZO 9400i	*20 KOLOR MONITOR*
EIZO T960	*20" KOLOR TRINITRON MONITOR*
EIZO MD B08	*GRAFIČKA KARTICA*
EIZO MD B10	*GRAFIČKA KARTICA*
EIZO MD B12—60	*GRAFIČKA KARTICA*

ISPORUKA ODMAH
GARANCIJA GODINU DANA
POSEDUJEMO DODATNI I POTROŠNI MATERIJAL
OVLAŠCENI SERVIS
INFORMACIJE NA TEL 011 434 812
FAX 011 450 471

Vi niste dovoljno bogati da kupujete jeftino

Mi nudimo
telefax aparate
telefax papir
telefonske centrale sa 3, 8, 16 ulaza
fotokopir mašine
šredere
laserske i matrične štampače
DTP monitore
računarske mreže

COMPAQ, EVEREX, HP, PANASONIC
WYSE, EPSON, D-LINK

OBNAVLJANJE TONERA LASERSKIH KASETA
Kompletan asortiman „OLIMPIJA"

USA aparati/ USA cene
Garantovane cene nakon uplate. Do 3 godine garancije.

MICRO ENGINEERING INTERNATIONAL

Makedonska 34, 11000 Beograd, Jugoslavija
Tel: 011 328-051, 324-505, 328-526
Fax: 0038 11 453-628, 328-441

YUCO PRINT

Poklon svakom kupcu knjige
VENTURA 3.0 Professional
YUCO fontovi

Knjiga koju ste nestrpljivo očekivali

VENTURA 3.0 Professional

Detaljan opis SVIH funkcija VENTURE
- rad sa tabelama
- rad sa formulama

Specijalan poklon

YUCO fontovi za VENTURU

YUCO-PRINT 21000 Novi Sad Todora Jovanovića 14
Tel 021/300-288 Fax 021/396-424

N A R U D Ž B E N I C A

Ovim neopozivo naručujem _____ primeraka knjige "VENTURA 3.0 Professional" po ceni od 390 din po primerku.

Prezime i ime _____
Ulica i kućni broj _____
Mesto _____ Telefon _____
Troškovi poštarine padaju na teret primaoca.

HSM INFORMATIKA

Može li laserski štampač zamijeniti fotoslog

Mnogi se od Vas grafičara izdavača knjiga i novina dizajnera sitotiskara i grafičkih urednika sa sumnjom pitaju da li se laserski štampac uopće može koristiti u profesionalnoj pripremi za tisak.

Otisak laserskog štampača s 300 tpi zadovoljava samo amatera

Medutim LaserMaster laserski štampaci kvalitete otiska od 400 pa sve do 1200 tpi pokazuju grafički crtež fantastične oštrine na svim uglovima. Linije i krivulje slova se štampaju oštre cisle i bez zubaca tako da LaserMaster štampači mogu izdržati usporedbu sa mnogo skupljim i nepraktičnijim fotoslogom.

Mi smo preuzeli teret Vaše odluke

I napravili za korisnike različitih zahtjeva različita DTP rješenja koristeći isključivo komponente najpoznatijih svjetskih proizvodaca hardvera LaserMaster Canon Hewlett Packard i Wyse. Od softvera isporucujemo Venturu, PageMaker Corel Draw Picture Publisher i Wordperfect

Iz paketa s 135 fontova od kojih svi sadrže naše znakove koje isporučujemo našim korisnicima navest cemo samo neke: **Sans (kao Helvetika)** **New Serif (kao Times Roman)** Courier **Hobo** *Mariage* *Mummy Stilt*

LM EUROPE LASERMASTER CORPORATION

distributer za Jugoslaviju

Heroegovačka 9, 41000 Zagreb, tel: (041) 570-251, fax (041) 573-080

INSTITUT ZA NUKLEARNE NAUKE "BORIS KIORIĆ", VINČA
CENTAR ZA PERMANENTNO OBRAZOVANJE
11000 Beograd, Nemanjina 4/X
Telefoni: (011) 683-390, 682-486, 641-155/107,181
Telefax: (011)682-486

KNJIGE U OBLASTI PROGRAMIRANJA, PROGRAMSKIH JEZIKA I PRIMENE RAČUNARA

1. AutoCAD (verzija 10.0)
konstruisanje i projektovanje pomoću personalnih računara
(šesto izdanje, 1991.)
Autori: Boris Damjanović i Petar Damjanović
Latinica, 444 strane, format B-5, cena: 780 dinara

2. Uvod u C jezik
(treće izdanje, 1990.)
Autor: Vladan Vujičić
Latinica, 317 strana, format B-5, cena: 585 dinara

3. Primena programa SYMPHONY na personalnim računarima
(treće izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 226 strana, format B-5, cena: 455 dinara

4. OS/2 - vodič za korisnike
(prvo izdanje, 1989.)
Autor: Zorica Jelić
Latinica, 253 strane, format B-5, cena: 455 dinara

5. VENTURA - računarsko izdavaštvo
(treće izdanje, 1991.)
Autor: Predrag Davidović
Latinica, 253 strane, format B-5, izlazi iz štampe u oktobru '91

6. FORTRAN 77
standard sa dopunama za personalne računare
(drugo izdanje, 1990.)
Autori: Vlajko Kocić i Zoran Konstantinović
Latinica, 422 strane, format B-5, cena: 780 dinara

7. UNIX - vodič za korisnike
(drugo izdanje, 1990.)
Autor: Zorica Jelić
Latinica, 422 strane, format B-5, cena: 780 dinara

8. Primena programa FRAMEWORK III na personalnim računarima
(prvo izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 326 strana, format B-5, cena: 585 dinara

9. PROGRAMSKI ALATI U MATEMATICI
MathCAD, Grapher, Eureka
(prvo izdanje, 1990.)
Autor: Ante Čarlin
Latinica, 402 strane, format B-5, cena: 715 dinara

10. Primena programa QUATTRO na personalnim računarima
(prvo izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 296 strana, format B-5, cena: 585 dinara

11. DOS ukratko
(prvo izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 89 strana, format B-5, cena: 260 dinara

12. Vodič za VAX/VMS
(prvo izdanje, 1990.)
Autori: Tamaš Kerepeš, Zvonko Orešolić, Saša Matijević
Latinica, 512 strana, format B-5, cena: 910 dinara

13. Primena programa EXCEL na personalnim računarima
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 272 strane, format B-5, cena: 585 dinara

14. UNIX - vodič za programere
(prvo izdanje, 1991.)
Autor: Zorica Jelić
Latinica, 326 strana, format B-5, cena: 715 dinara

15. WINOOWS 3.0
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 273 strane, format B-5, cena: 585 dinara

16. PRIMAVERA - upravljanje projektima uz pomoć računara
(prvo izdanje, 1991.)
Autori: Jaroslav Urošević i Jelica Drašković-Ostojić
Latinica, 365 strana, format B-5, cena: 780 dinara

17. dBASE III+ priručnik
(prvo izdanje, 1991.)
Autor: Milorad Filipović
Latinica, 249 strana, format B-5, cena: 559 dinara

18. Osnovi informaciologije i informacione tehnologije
(prvo izdanje, 1991.)
Autor: Ljubomir Dulović
Latinica, 538 strana, format B-5, cena: 1001 dinar

19. LOTUS 1-2-3 (verzije 3.0 i 3.1)
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 411 strana, format B-5, izlazi iz štampe u septembru '91

20. dBASE IV priručnik
(prvo izdanje, 1991.)
Autor: Ljubomir Lazić
Latinica, 306 strana, format B-5, izlazi iz štampe u septembru '91

"RAČUNARI", septembar 1991.

Poručujem (ispod rednog broja knjige upisati broj poručenih primeraka)

Redni broj knjige	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
Broj poručenih primeraka																				

IME i PREZIME
(Naziv preduzeća) _____

ULICA i BROJ _____

BROJ POŠTE i MESTO _____ **TELEFON** _____

Poručbenicu poslati sa petim primerkom uplatnice na adresu:
Institut za nuklearne nauke "Boris Kidrič", Vinča
Centar za permanentno obrazovanje, Beograd, Nemanjina 4/X.
Uplata se vrši na žiro račun:
INSTITUT ZA HEMIJSKU DINAMIKU I PERMANENTNO OBRAZOVANJE, broj: 60803-603-17361.
Troškove poreza na promet i isporuke knjiga snosi poručilac i namiruje ih unapred zajedno sa uplatom za knjige

Computer hit bibhoteka
Poštanski fah 116 71210 Ilidža
Tel 071/ 270-213 i 071/621-025

Computer hit biblioteka
LITERATURA ZA IBM PC

Novo u izdanju Computer hit biblioteke
Edicija "Džepna knjiga za PC"

1. MS DOS 3.3/4.0

Knjiga neophodna svim korisnicima IBM PC i kompatibilnih računara. Detaljan opis komandi sa svim opcijama koje su na raspolaganju za pojedine verzije MS DOS a. Objašnjenja o primjeni. Posebna napomena za verziju 4.0

125 strana 230.00 din

2. PC Tools Deluxe 6.0

Priručnik za rad sa najpopularnijim pomoćnim programom za personalne računare. Knjiga sadrži opis instaliranja programa korisničkog interfejsa, rada sa datotekama i direktorijumima kao i objašnjenje za rad sa programima koje sadrži paket PC Tools: PC Shell, PC Backup, Compress, Diskfix, PC Cache, Desktop

130 strana 230.00 din

3. WordPerfect 5.1

Detaljan podsjetnik za rad sa tekst procesorom koji je osvojio svijet. Sve opcije programa su date abecednim redom što omogućava njihovo lako snalaženje. Opisuje rada korak po korak. Nove mogućnost WordPerfecta: tabele, formule, stilovi i makro naredbe

130 strana 230,00 din

4. MS Quick Basic 4.5

Referentni priručnik za popularni programski jezik. Opis svih instrukcija i funkcija sa tačnom sintaksom i načinom pozivanja. Ilustrovano mnogobrojnim primjerima

140 strana 230.00 din

5. Norton Utilities

Priručnik za rad sa svim komponentama sistema Norton Utilities od instaliranja programa i testiranja vašeg računara do korištenja naprednih mogućnosti ovog vrlo korisnog programa

120 strana 230,00 din

6. Lotus 123 Rel. 2.2/3.0

Podsjetnik za rad sa integrisanim paketom programa koji je nametnuo svjetski standard. Kompletan pregled svih komandi, opcija, funkcija i makroa programa.

150 strana 230.00 din

7. Biblioteka Runtime funkcija za MS C

Knjiga koja će vam olakšati rad i uštedjeti trud i vrijeme za razvoj najčešće korištenih funkcija. Jednostavno neophodna svakom C programeru. Detaljan opis više od 300 funkcija

250 strana 330.00 din

8. MS Windows 3.0 *Windows verzija!*

Operativni sistem nove generacije. Budite među prvim koji će ovladati njegovim ogromnim mogućnostima

160 strana 250.00 din

9. Clipper 5.0

Pregled svih instrukcija, komandi i funkcija sa sintaksom, namjenom, opisom argumenata i primjerom korištenja. Podešavanje okruženja programa, instrukcije pretprocesora, standardne klase, funkcije extend sistema

320 strana 390.00 din

10. AutoCad 10

Referentni priručnik za rad sa popularnim programom namijenjenim dizajniranju, projektovanju, uz pomoć računara u kom se na sistematičan način rade objašnjenje o radu sa svim komandama programa i raspoloživim opcijama.

280 strana 390,00 din

11. Ventura Publisher 3.0

Referentni priručnik za najpopularniji program iz oblasti stonog izdavaštva (DTP) na PC u. Instaliranje programa, sistematizovan pregled svih opcija, opis rada sa tabelama i formulama, instalacija fontova. Obrađivane su verzije za GEM i MS Windows

250 strana 390.00 din

12. CorelDRAW! 2.0 *Windows verzija!*

Upoznajte se sa fascinantnim karakterističkama najboljeg programa za crtanje i dizajniranje na PC računarima. U novoj verziji donosi rad u 3D perspektivi, ovojnice, nove fontove, mogućnost kreiranja sopstvenih fontova, konverziju fontova u i iz raznih formata, novi tracing program i još puno toga

140 strana 230.00 din

U pretplati (izlazi iz štampe 25.06.1991.)

13. QuattroPro 2.0

Referentni priručnik za rad sa jednim od najboljih programa za tabelarne proračune. Dat je pregled komandi, funkcija i makroa i opisan rad sa grafikom. Ako vas nalite sa programom QuattroPro ove knjiga će vam biti nezamjenjiv pomoćnik

140 strana 230.00 din

14. Word for Windows 1.1 *Windows verzija!*

Upoznajte tekst procesor nove generacije. Iskoristite prednosti grafičke radne okoline WYSIWYG pri kazi i štampanja, dinamičke razmjene podataka sa ostalim Windows aplikacijama, moćnih makroa i ostalih prednosti ovog izvanrednog programa

140 strana 230.00 din

U pretplati (izlazi iz štampe 15.07.1991.)

15. MS Excel 3.0 *Windows verzija!*

Ova knjiga će vas uvesti u sve tajne Excela, trenutno najmoćnijeg programa za tabelarne proračune. Jasan pravi izbor za korisnike koji traže snagu ali i fleksibilnost i lakoću upotrebe. Pored osnova rada, komande, funkcije i makroa programa, čak je i prikaz poređenih alata Solver i Q+E

150 strana 250.00 din

Knjige možete naručiti poštom – priloženom narudžbenicom ili telefonom na brojeve
071/270-213 i 071/621-025 od 09-17 sati

Computer hit biblioteke Poštanski fah 116 71210 Ilidža

N A R U D Ž B E N I C A

Naručujem sljedeće naslove iz edicija "Džepna knjiga za PC"
(upišite broj kumada ispod odgovarajućeg broja knjige)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Prezime i ime _____

Adresa _____

Mjesto _____ Telefon _____ Fax _____

Tačnost potpune i pokomplet podaju na teret naručioca

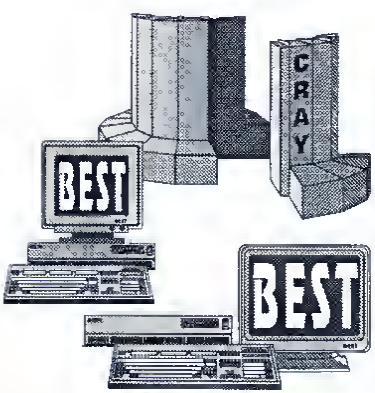
BEST COMPUTERS

011 320-103

BEOGRAD, MAJKE JEVROSIMA

PC AT 2B6	16 Mhz - 1 MB RAM		40.900,-
PC AT 2B6	16 Mhz - 2 MB RAM		43.900,-
PC AT 2B6	20 Mhz - 1 MB RAM		43.900,-
PC AT 3B6 SX	16 Mhz - 2 MB RAM		54.900,-
PC AT 3B6	25 Mhz - 4 MB RAM		76.900,-
PC AT 3B6	33 Mhz - 4 MB RAM		83.900,-
PC AT 4B6	25 Mhz - 4 MB RAM	64 KB CACHE	117.900,-
PC AT 4B6	33 Mhz - 4 MB RAM	128 KB CACHE	124.900,-

STANDARDNO: 42 MB HD * 1,2 MB FD * HERCULES
* AT BUS CTRL * MONO MONITOR * TASTATURA *
DOPLATA ZA VGA KOLOR 22.000,- VGA MONO 17.000,-
GARANCIJA 12 MESECI * NON - STOP SERVIS



POSTOJI SAMO JEDNA MOGUĆNOST

BEST

CTT DESIGN 011 463-655

Microline

Sjedište Jordanovac 119 Zagreb prodaja i administracija, Kraljevićeva 18a, tel 041/232-636, 041/232-119, fax 041/216-711 mobitel 099/410-267

skladište Šenoova 25 tel 041/217 915 servis Microline-Tekom, Mekrička 13 mobitel 099/410-264

Radno vrijeme 8-17 (pon-čet) 8-16 (petak)

Žiro račun 30105-601-35909

Kompjuteri

Microline AT 16/40

Takt 16MHz, RAM 1 MB, hard disk 40 MB 28 ms, floppy 1.2 i 1.44 MB, hercules

42.000,-

Microline 386SX 16/50

Takt 16 MHz, RAM 1 MB, hard disk 52 MB 17 ms, floppy 1.2 i 1.44 MB, hercules

57.800,-

Doplata za 20 MHz 1.800,-

Microline 386 25/100

Takt 25 MHz, RAM 4 MB, hard disk 100 MB 17 ms, floppy 1.2 i 1.44 MB, hercules, mini tower

98 500,-

Microline 386 33/100

Takt 33 MHz, 64 KB cache RAM 4 MB, hard disk 100 MB 17 ms, floppy 1.2 MB hercules mini tower

106 700,-

Microline 486 EISA

Takt 33 MHz, EISA arhitektura, RAM 16 MB, hard disk 320 MB 13 ms, EISA hard disk kontrolor, floppy 1.2 MB hercules tower kućište

319 000,-

U kompjutere ugrađujemo hard diskove Conner, Quantum, IBM i Maxtor, floppy fujitsu TEAC i Mitsubishi, tastature Cherry, sve hercules ima serijski i paralelni port te YU set

Doplate za opcije

Umjesto hercules VGA kartica 1 MB s VGA monitorom u boji Tyster (1024*768)

21.300,-

Umjesto hercules VGA kartica 1 MB s multisync monitorom u boji EIZO 9070S (1024*768 u 256 boja, 14")

55.000,-

Umjesto hercules kartica 1280*1024 (EIZO MD 1280) i monitor 20" u boji (EIZO 9400i)

197.000,-

Umjesto hercules VGA kartica 1 MB s monokromatskim monitorom 720*480 (EIZO 4050)

11 700,-

RAM 2 MB umjesto 1 MB	2 300
RAM 4 MB umjesto 1 MB	7 300
RAM 9 MB umjesto 4 MB	11 000
RAM 16 MB umjesto 4 MB	40 000
RAM 32 MB umjesto 4 MB	132 000
RAM 64 MB umjesto 4 MB	286 000
Disk 40 M u 50 M 17 ms	1 800
Disk 40 M u 80 M 19 ms	7 700
Disk 40 M u 105 M 17 ms	11 000
Disk 105 M u 170 M 15 ms	14 500
Disk 105 M u 210 M 15 ms	19 200
Disk 105 M u 320 M 13 ms	52 000
Disk 105 M u 400 M 11 ms	63 000
Disk 105 M u 1 G 14 5 ms	137 000
Dodatni floppy 1 44M	4 000

Mreže



Kompjuteri spojeni u mrežu daju kapacitivan sistem pomoću kojeg je moguće voditi poslovanje i velikih poduzeća bez kompliciranja usluga velikih računala.

Novell software	cijene na upit
Novell hardware	11 000 /čvor
Novell hardware	20 000 /server

U cijene je uključeno sav potrebni hardware i instalacija mreže

Miševi

GM 6	1 300
3 tipke, 2S+ priključak, kondektor	
GM F302	2 600
1050 dpi, ergonomična raspodjela, 3 tipke, podloga i držač za miša, software	

Modemi

Modem 2400 bauda	5 500
MVoice+ software + kabel (US)	

Kućišta

AT flip top + 200W	5 100
AT slim line + 200W	5 700
Mini tower + 200W	7 500
Tower + 220W	12 100

Sva kućišta osim flip top imaju LED za takt i mjesta bar za po 4 disk jedinice

Tastature

Tastatura 101 tipka	2 300
Tastatura 108 tipki	2 750

PC Magazine Editor's Choice, Quality Product Award, PCO Aprovec

Monitori

Hercules monitor	5 400
VGA 1024*768	19 800
Tysinr, 14", interlaced	
EIZO 4050	11 600
Monokromatski VGA 720*540, 14"	

Čipovi

RAM

41256-80	85
44256-80	330
1Mx1-80	350
SIMM 256/1-80	1 040
SIP 256K-80	1 000
SIP & SIMM 1M-80	3 300

Kaprocesori

IIT 2C87-10	6 200
IIT 2C87-12	6 800
IIT 2C87-20	8 400
IIT 3C87SX-16	8 900
IIT 3C87SX-20	10 900
IIT 3C87-20	13 200
IIT 3C87-25	14 300
IIT 3C87-33	17 000

IIT koprocesori su dva puta brži od Intela, duvrstveni su i ugrađeni u njem kompatibilni. Omogućavaju direktnu zamjenu Intelovih čipova.

Cyrix 83D87SX 16	10.500
Cyrix 83D87SX-20	11.700
Cyrix 83D87-20	14.300
Cyrix 83D87-25	15.400
Cyrix 83D87-33	17.000

Cyrix koprocesori su 5 puta brži od Intela

Weitek 3167-25	31 700
Weitek 3167-33	40.000
Weitek 4167-25	40 000
Weitek 4167-33	52.000

EPROMi i ROMovi

2764-250	121
27C128-150	135
27C256-150	135
27C512-150	250
ROM YU set za hercules	180

Eksterna memorija

Hard diskovi

Maxt. ili WD 40M 28 ms, AT	10 500
Quantum 50M 17 ms AT	13 500
M tsuhishi 64M 28 ms RLL	17 200
Quantum 80M 19 ms AT	20 300
Quantum 105M 19 ms AT	23 400
Quant. 104M 19 ms SCSI	24 200
Quantum 170M 15 ms AT	36.500
Quantum 170M 15 ms SCSI	36.800
Quantum 210M 15 ms, AT	42.000
Quant. 210M 15 ms SCSI	42.300

Quantum diskovi imaju 2 godine garancije, npr 50 000 h MTBF i ext su u 5 ¼

IBM 320 M 13 ms, SCSI	59 400
IBM 400 M 11 ms SCSI	69 000
Rame za hard disk 3 ½	155

Floppy diskovi

1.2 MB	4 100
1.44 MB	4 000

Streameri

Wangtek 40/120 MB	19 500
Wangtek 80/240 MB	25 000
Wangtek 80 MB s kontr	35 200
Wangtek 150/500 MB e kon.	43 000
Wangtek 525 MB SCSI	43 200
Wangtek 1 3 GB	64 300
Traka DC2120	2 000
Traka DC6150	2 100

Monitori

EIZO 9060S	36 600
VGA u boji 800*600, 14"	
EIZO 9070S	52 000
VGA u boji 1024*768 16"	
EIZO 9400i	122 600
1280*1024, 20", boja	

Diskete

5.25" DD 10 kom	520
5.25" HD 10 kom	630
3 5" DD 10 kom	550
3.5" HD 10 kom	850

Diskete su BASF, 3M ili Maxell

Printeri

Epson LX400	11.700
A4, 9 iglica, uskiri kolica, 180 cps	
Epson LX850	18.000
A3, 9 iglica, uskiri kolica, 200 cps	
Epson LX1050	23.300
A3, 9 iglica, 200 cps	
Epson FX1050	28.800
A3, 9 iglica, 300 cps	
Epson FX850	20 700
A4, 9 iglica, 300 cps	
Epson LQ550	24.200
A4, 24 iglice, 180 cps	
Epson LQ 50+	40.500
A4, 24 iglice, 264 cps, 30-K buffer	
Epson LQ560	nazovite
A4, 24 iglice, 300 cps, boja	
Epson LQ1010	27.800
A3, 24 iglice, 180 cps	
Epson LQ1050+	47 300
A3, 24 iglice, 264 cps, 30K buffer	
Epson LQ1060	nazovite
A3, 24 iglice, 300 cps, boja	
Epson LQ2550	nazovite
A3, 24 iglice, 400 cps, boja	
Epson DFX5000	nazovite
A3, 9 iglica, 533 cps, heavy duty	
Epson DFX8000	nazovite
A3, 9 iglica, 1095 cps, heavy duty	
CSF LX400, LQ400	4 400
CSF LQ850, FX850	9 300
CSF LQ1050, FX1050	11 200
CSF LQ2550, dvostruko	22 300
CSF LQ550, LX800	5 300
CSF LQ800	10 400
CSF LQ1060	12 300
CSF LX1050, LQ1010	7 100
CSF + automatski uvlakač papira	
IS 232 za Epson printere	2 500
HP IIP laser	70 000
HP IIP laser	103 000
HP IID laser	157 000
HP IIISi laser	243 000
HP Painjet	60 000
Tonei za HP IIP	4 700
Tonei za HP IVIII	5 500
Pacific Data PostScript	21 500
Hardvarski YU set za HP	7 500
RAM kartica za HP, 0KB	5 200
RAM 1 MB za HP	7 200
RAM 2 MB za HP	9 200
RAM 4 MB za HP	13 200
Kabel za printer	340
Traka FX800/850, LX400/500	275
Tr FX105/1000/1050, LX1050	385
Traka DFX5000/8000	1 675
Traka LQ2550/860/1060	540
Traka LQ400/550/650	400
Traka LQ 1010/1050	1 040

Sve su trake originalne Epson

Skeneri

Epson GT-4000	85 600
Epson GT-6000	96 000
Epson GT 1000	36 400

GT 6000 i 5000 su skeneri u boji

HP ScanJet+	97 000

Nepreki dni izvori napajanja

450 VA	15 000

Ploteri

Roland DXY 1100	45 000
Roland DXY-1200	59 000
Roland DXY-1300	83.000

DXY-1100, 1200 i 1300 su A3 formata

Roland DPX 2500	239 000

A2 format, ploter-tablia

Roland DPX 3500	266 000

A1 format

Roland DPX-4600	455 000
Roland GRX 300	243 000
Roland GRX-400	305 000
Roland LTX 100	80 000
Roland LTX 120	157 000
Roland LTX 321	468 000
Roland LTX 420	540 000

LTX su termički ploteri

Roland CAMM 1 PNC 1000	146 000

Ploter-rezač

Roland CAMM 1 PNC 1600	480 000
Roland Stika	32 000

Scanner-rezač

Roland Sketch Master	29 000

Ploter-rezač

Literatura

Sistemske progr za IBM PC/PS2 s25

Štampačke ploče

Matične ploče

AT 16 MHz	6.500

2/3 baby veličine, do 5 MB, EIM EMS 4.0

386SX 16 MHz	19 600
386SX 20 MHz	20 300
386 25 MHz	34 300
386 33 MHz, 64K cache	42.900

Kontroleri

AT bus	1.320
AT bus + I/O	1 600
SCSI WD7000 FASST	14 000
ESDI WD1007SE2	8 100

Grafičke kartice

Hercules YU	1.100
VGA 1MB 1*ident	6 500

1024*768, 256*1024, drivers za Windows 3 0, Presentation Manager, Autocad, Ventura, GEM

I/O kartice

AT I/O S+P+G	860
IEEE 488	7 900
UNIX 8 port, Chase R.	28 600

Ethernet

Ethernet kartica, 8 bit	5 300
Ethernet kartica, 16 bit	7 000
Western Dig tal 8 bit	7 700
Western Dig tal, 16 bit	8 500

Garancija 12 mjeseci Cijene važe za poduzeća i u njih nije uračunat porez na promet. Sve cijene su iz Zagreb Šenoova 25 ali je dostava računala u Zagrebu besplatna. Organiziramo ef kasnu dostavu u sve republike željezničkom poštom, YUPD-om i avionom Robu šaljemo sa dan po dospjelo uplate na naš žiro račun Plaćanje isključivo 100% avansom. Ovaj cjenik vrijedi za službeni kurs 1DEM=13DIN. Cijene bilo kada ovise o mogućnosti plaćanja u inozemstvu. Za trenutne cijene nazovite Distributeri Housing Ljubljana, tel 061/621 145 i JWare Osijek, tel 054/603-459

MLAKAR & CO
AUSTRIJA

Računare prodajemo u KIT verziji (u delovima). Za sve uređaje nudimo garanciju, a montažu i servis u Jugoslaviji. Za savete kod izbora pozovite nas na telefon: 9943/4227 2333. Naša trgovina je u Austriji, u Podgori (Unterbergen), kraj glavnog puta prema Celovcu, 60 km od Ljubljane i 12 km od Ljubelja. Trgovine je otvorena od 8 do 17 časova, a u subotu od 8 do 13 časova.

FAKS 9943/4227 2091 TELEKS 422749 MLCO A



IZUZETNA PONUDA
RAČUNAR + ŠTAMPAČ

PC-M 286-12-45 S ŠTAMPAČEM
– AT 286/12 MHz, 1 Mb RAM, japanski brzi disk 45 Mb, FDD 5,25" 1,2 Mb (ili 3,5" 1,44 Mb), tastatura 101 14", monohromatski monitor, štampač Citizen 9 igl. A4
DIN 48.800
– s štampačem formata A3
DIN 60.200

PC-M 286-16-45-NEAT S ŠTAMPAČEM
– AT 286/16 NEAT MHz, 1 Mb RAM, japanski brzi disk 45 Mb, FDD 5,25" 1,2 Mb (ili 3,5" 1,44 Mb), tastatura 101 14", monohromatski monitor, štampač Citizen 9 igl. A4
DIN 50.100
– s štampačem formata A3
DIN 61.400

PC-M 386-SX-45 S ŠTAMPAČEM
– AT 386 SX/16 MHz, SX 1 Mb RAM, japanski brzi disk 45 Mb, FDD 5,25" 1,2 Mb (ili 3,5" 1,44 Mb), tastatura 101 14", monohromatski monitor, štampač Citizen 9 igl. A4
DIN 60.600
– s štampačem formata A3
DIN 71.800

Garancija 24 meseci

MLACOM d.o.o.
Celovška 185
66100 Ljubljana 1

KUĆIŠTA SA NAPAJANJEM	DEM	BODOVI
AT BABY	117	187,20
SLIM	154	246,40
MINI TOWER	187	299,20
TOWER	255	408,10
☆ FILE SERVER 375W	950	1.520,00
WORKSTATION	192	308,80

OSNOVNE PLOČE		
XT 4,77-10 MHz	115	184,30
☆ AT 286-12MHz	145	203,20
☆ HEADLAND 286-16MHz	185	287,20
☆ HEADAK 286-20	235	376,00
386 SX-16 MHz	540	922,44
☆ 386 SX-16 MHz ALL-IN-ONE	390	712,00
☆ 386-SX 20 MHz	590	792,00
☆ 386 25MHz CACHE	1 290	1 368,00
386 33MHz CACHE	1 580	1 758,40
486-25 MHz	3 400	4 792,00
486-25 MHz, 64 KB CACHE	3.600	4.864,00
486 25 MHz, 64KB CACHE EISA	5 271	9 434,00
486-33 MHz, 128 KB CACHE INTEL	3.900	5 104,00

DISPLAY KARTICE		
Printer Hercules	28	44,80
Printer/Hercules/CGA	39	62,80
VGA 800X600/16 bit	124	144,00
Super VGA 1024x768	160	233,00
GENOA 8200 SUPER VGA	255	408,00
GENOA 8300 SUPER VGA	266	425,10
GENOA 8400 SUPER VGA	420	672,00
GENOA 8400 A SUPER VGA	570	912,00
GENOA 8400 V SUPER VGA	495	792,00
GENOA 8200 VC SUPER VGA	660	1 056,00
VGA 1280 × 1024 (NEC 5D)	3.427	5 483,00

KONTROLERI		
HDD XT MFM	60	84
FDD/HDD AT MFM 1:1	70	84,80
DTC-7280 AT MFM 1:1	187	299,40
DTC-7287 AT RLL 1:1	197	315,40
ATIDE(BUS FDD/HDD	35	56,00
SCSI FDD/HDD	82	131,20
ESDI FDD/HDD	280	448,00

DODATNE KARTICE		
MULTI I/O XT	66	104,60
I/O AT (SER. PORT)	22	32,00
I/O AT (PAR/2×SER PORT)	30	48,00
I/O AT (PAR/2×SER GAME)	33	52,80
MULTI USER (4×RS232)	169	270,40
MULTI USER INTELLIG (8×RS232)	713	1 140,80
AD/DA 12 bits	137	219,40

LAN		
Ethernet compat. INE 1000) 8 bit	235	376,00
Ethernet compat. INE 2000) 8 16bit	280	448,00
Ethernet Pocket Adapter	618	965,10
Ethernet boot rom for NE 1000	10	22,60
Ethernet boot rom for NE 2000	10	22,60
IEEE802.3 transceiver	212	340,60
BNC 50 ohm terminator	5	8,10
BNC 93 ohm terminator	5	8,10
N series 50 ohm female terminator	9	13,70
Cable RG 58 (1M)	3	4,80
Cable connector	5	8,10
Ethernet IEEE802.3 repeater	1 207	1 931,40
Arcnet coax-star LAN card	125	201,10
Arcnet coax bus LAN card	138	221,70
Arcnet twisted pair star LAN card	138	221,70
4 port coaxial act ve hub card	378	502,80
4 port twisted pair hub card	378	249,10
Remote boot rom for arcnet card	10	16,50
Cable RG 62 (1M)	3	4,80

TASTATURE		
102 tipke	58	92,80
101 tipka click Cheeky YU	78	107,20
101 tipka s mišem Chicany	167	267,50
101 tipka Cheny	135	206,40

MEKI DISKOVI		
5,25" 360 Kb	111	177,60
5,25" 1,2 Mb	115	184,00
3,5" 1,44 Mb	115	184,00

TVRDI DISKOVI		
Seagate – 157A 45Mb/28ms	390	588,80
Seagate – 274A 83Mb/28ms	750	1.249,20
Seagate – 296A 85Mb/28ms	680	1 104,00
Seagate – 1102A 89Mb/18ms	710	1 136,00
Seagate – 1126N 107Mb/15ms	1 406	2 250,30
Seagate – 1144A 125Mb/18ms	990	1 584,00
Seagate – 2182E 160Mb/15ms	2 141	3 426,30
Seagate – 1201A 177Mb/15ms	1 499	2 397,70
Seagate – 2209N 183Mb/18 ms	2 140	3 424,00
Seagate – 1239A 211Mb/15ms	1 600	2 560,00
Seagate – 2383A 336Mb/16ms	2 640	4 224,00
Seagate – 2383E 338Mb/15ms	2 855	4 569,10
ST 2383N 337 Mb/14 MS	2 855	4 569,10
ST 2502N 442 Mb/16 MS	3 569	5 709,70
ST 4376NV 330 Mb/16 MS	3.000	4 800,00
ST 4385NV 337 Mb/10 MS	3 405	5 448,00
ST 2502NV 442 Mb/16 MS	3.735	5.976,00
ST 4766NV 676 Mb/15MS	3.945	6 312,00
ST 41200NV 1 050 Mb/15 MS	5 713	9 140,00
NEC		
D 3741 A 44 MB/23 MS	540	784,00
D3661 135 MB/23 MS	1 350	2 160,00
D 5655 179 Mb/18 MS	1 490	2.384,00

MONITORI	DEM	BODOVI
14" mono hromatski	175	272,00
VGA monohromatski	228	336,20
VGA Color 1024×768	598	880,80
15" A4 full size VGA	1.540	1 896,00
NEC Multisync 2A 3D 4D 5D		
VGA CITIZEN 14" 1024 × 768	1.300	1.920,00
PANASONIC 20" 1280 × 1024	4 427	7 083,40
MITSUBISHI VGA 720 × 400	975	1 560,00
MITSUBISHI 14" MULTISCAN	1.350	2 160,00
MITSUBISHI VGA 15" 1024 × 768	3 300	5 280,00
QUME VGA 14" 1024 × 768	1 071	1 714,00

ŠTAMPAČI		
1 CITIZEN 180D A4	286	454
2 C.T.I. 9 Pin A3	638	1 018,30
Star LC-20	405	648,00
Star LC 15	733	1 104,00
Star LC 24-200	698	1 054,00
Star LC-24 15	1 010	1 616,00
Star ostali modeli		
Laser HP JET III P		4 569,00
Laser HP JET III		8 060
Laser HP JET IIISi		18 240
CANON serija BJ		
QUME serija CRYSTAL PRINT		

CRTAČI		
ROLAND DXY 1100 A3	1 690	2 672,00
ROLAND DXY 1200 A3	2 068	3 341,70
ROLAND ostali modeli		

MODEMI		
2400 int.	193	292,90
2400 ext. (MNP5)	243	388,30
9600 ext. (MNP5)	1 081	1 577,10
2400 POCKET	227	363,40

UPS – NEPREKIDNO NAPAJANJE		
UPS 300 VA	480	756
UPS 500 VA	570	912
UPS 1000 VA	1 097	1 755

RAM		
41256-10	3	4,60
41256-08	3	4,80
44256-08	11	17,60
411000-08	10	16,00
SIMM/SIP 256K×9/08	35	56,00
SIMM/SIP 1ME×9/08	97	156,00

COPROCESSOR		
80287	278	728
80387SX 16MHz	650	1 040
80387 25MHz	790	1 648
80387 33MHz	1 080	1 760

STREAMER		
COLORADO 40/60/120 Mb int.	785	1 182
TARGA 150 Mb ext.	1 729	2765

RAZNO		
PC NOTEBOOK XT 20 Mb	1 990	3 184
PC NOTE BOOK 286 VGA 20 Mb	3 700	5 040
FAX N-SSEI	1 149	2 011
FAX CAPD	557	891,40
FAX MODEM CARD		
FAX MODEM POCKET		
Čitač bar koda	486	745,10
Prenosni čitač bar koda	1 042	1 667,70
CCD Scanner		
Štampač bar kode		
Miš Genius 6 Plus	68	108,90
Miš Genius GM F 302	87	139,20
Miš bežični	166	265,70
Track Ball	86	104,90
Tablet Genius GT 1212E 12×12	533	851,20
Tablet Genius GT 1612D	1 027	1 589,90
Scanner Handy Genscan GS-4500	290	456,00
Scanner A4 Handy wpap feeder	1 120	1 792,00
Scanner EPSON GT-6000 Color	2 850	4 560,00
Eprom UV Eraser	280	418,00
Eprom Writer Card 4×	392	626,00
Disk Box 5×5,25"	2	3,40
Disk Box 10×5,25"	4	5,70
Disk Box 50×5,25"	12	19,20
Disk Box 5×3,5"	3	4,80
Disk Box 10×3,5"	3	4,80
Copy Holder	14	22,80
Pločnavač za monitor i tastaturu	19	30,20
Sve vrste EPROM		

DEM su cene bez poreza na promet kod Mlakar & CO Austrija

Cene kod Mlacoma su bez poreza na promet – 1 bod vrednost DEM na Ljubljanskoj berzi

Na zalihi i druga oprema

Tel 061/193-010 Fax 061/193-016
061/553-276

MLAKAR & CO

AUSTRIJA

Posebna ponuda

LIČNI RAČUNAR 286-16 SA ŠTAMPAČEM
286-16 MHz HEADLAND, 1 MB RAM, 1,2 MB meki disk,
40 MB tvrdi disk, monohromatski monitor,
štampač A4 Citizen 180D
DM 1372.-
Ista konfiguracija sa štampačem A3 C.T.I.
DM 1525.-

LIČNI RAČUNAR 386SX-16 SA ŠTAMPAČEM
386SX-16 MHz, 1 MB RAM, 1,2 MB meki disk,
40 MB tvrdi disk, monohromatski monitor,
štampač A4 Citizen 180D
DM 1781.-
Ista konfiguracija sa štampačem A3 C.T.I.
DM 1934.-

Za oba računara doplata za VGA 1024 × 706 kolor monitor i VGA kartica **DM 400.-**

MLAKAR & CO

Export-Import
Unterbergen 82
A-9163 UNTERBERGEN | Ros
AUSTRIA

telefon 9943-4227-2333
telefax 9943-4227-2091


ScanJet
LaserJet
IIIP
IIID
IIISI
III
HEWLETT PACKARD ASORTIMAN
Kalkulatori
LATINIČNI I ĆIRILIČNI FONTOVI
BITMAPIRANI I VEKTORSKI ISCRIPT
GRAFIČKI PROGRAMI
FONT KAZETE
U BOJI
PUNJENJE TONER KAZETA
CRNO
ERGO ARM nosač podlaktice
STALCI ZA PRINTERE OD PLEKSIGLASA
COPY HOLDERI
PERI HARD®
YUGOSLAVIA
KUTIJE ZA DISKETE
ZA ARHIVIRANJE
ZA SLANJE POŠTOM
FOOT EASE nosač stopala
PODUZEĆE ZADOVOLJNIH KORISNIKA
Klekovačka 28/I, pp 5030, Zagreb
tel: 041/263-696, fax: 041/263-714
STAR LIGHT
OPTIK GLAS
STAKLENI
FORTUNA
UP 'N DOWN
PROTECTOR
SUNFLEX
MIKROVLAKNA
FLEX
VEC
PROFESIONALNI ZAŠTITNI FILTERI
OPUS, Kranj
Tel (064) 36-785, 324-039 Fax (064) 325-879
FENIX, Novo Mesto
Tel (068) 28-126 Fax (068) 25-145
PROTEUS Sarajevo
Tel/Fax 071-422k-519
CODEX, Titograd
Tel/Fax (081) 24-149
EKOM, Banja Luka
Tel/Fax (078) 30-761
PC POOL, Sarajevo
Tel (071) 219-175 Fax (071) 31-764
ALMIRO, Skopje
Tel (052) 32-157, Fax (068) 562-975
S.C.T. Čačak
Tel (032) 41-592 Fax (032) 21-979
LTS, Skopje
Tel/Fax (091) 250-332
DI STRI RUTERI
ELEKTRODATA, Beograd
Tel (011) 340-016 Fax (011) 770-012
PERIHARD INTERNACIONAL Novi Sad
Tel. (021) 332 449
DEDRA, Osijek,
Tel (054) 124-166 Fax (054) 124-555
INFO D, Beograd,
Tel (011) 615-704

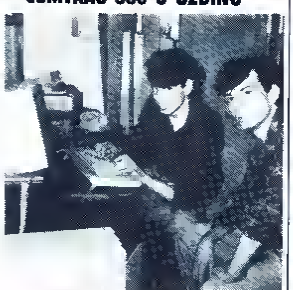
SERVIS KOMPJUTERA
– PC Commodore 64 Spectrum
– popravka diskova kasetaša ispravljača džojstika itd Tel 011/434 287
Avalska 6 Beograd

DISKETE
svih vrsta NASHUA, BASF računare i djelove sa garancijom prodajem po najpovoljnijim cjenah od 22 dinara dalje. Popust. Tel: (061) 265 525

PO NAJPOVOLJNIJIM cenama diskete kutije za diskete i printeri Star i Epson (069)22 216 ili 26 329

U prodaji je knjiga
PC MATLAB
sa elementima DOS-a
autori
Dr Ljubiša Stanković
Dr Zdravko Uskoković
cijena 398din strana 208
adresa "EPSILON",
Tehnicka fakulteti,
Cetinjski put bb Titograd
tel. 081-12-498

COMTRAO 386 U UZDINU



Najjaci računar Banata kako se nasalio jedan od članova naše redakcije, odskora se nalazi u Uzdinu. Kao što ste saznali iz prošlog broja „Računara", nešme dobitnik prve nagrade u našoj nagradnoj igri/anketi — računara „Comtrad" 386/33 MHz bogate konfiguracije — postao je Knstian Miklea iz tog malog banetskog mesta

Na dodelu nagrade u našoj redakciji Knstian (desno na slici) došao je „u dupikatu", sa bratom-blizancem Manuelom. Obojica su 21-godišnjaci odlični studenti elektrotehnike (završavaju drugu godinu), dugogodišnji privrženici računarstva (imaju „atari ST"), sa ozbiljnim životnim planovima u pogledu korišćenja računara „Comtrad" 386 je, dakle, otišao u prave ruke

Novi računar je na najpotpuniji način ispunio veliku želju brace Miklea da nabave 386. Sada ga koriste za programiranje, obradu teksta, obradu podataka, crtanje — dakle, sve same ozbiljne stvari — mada poneked odvoje malo vremena i za igre

E.J.

GOAP-gostol

servis

Naši ovlašćeni servisi su u Subotici Vinkovcima Slavonskom Brodu Sisku Bjelovaru Bihaću Varaždinu Zaboku Zagrebu Karlovcu Splitu Zadru Rijeci Bojanu Slovenskoj Bistrici Šmarje-Sapu Ljubljani Ajdovščini i Novoj Gorici

proizvodi i usluge

Iz naseg programa nudimo vam po povoljnim cenama kompletne računare

GOAP PC AT 286-12/H20-6/M	42 x00 – din
– GOAP PC AT 286-12W/H40-28M	47 900 – din
– GOAP PC AT 286-16W/H40-28M	49 800 – din
– GOAP PC AT 386-16S/H40-28M	59 800 – din
– GOAP PC AT 386-25/H40-19M	83 400 – din
– GOAP PC AT 386-25C64/H40-19M	93 900 – din
– GOAP PC AT 386-33C64/H40-19M	119 800 – din
– GOAP PC AT 486-25/H40-19M	218 700 – din

Za plaćanje u devizama dajemo dodatni popust

Cene su fco kupac



...postoje problemi koje može da reši samo

GOAP-GOSTOL, AUTOMATIZACIJA PROCESA, PRVOMAJSKA 124, 65000 NOVA GORICA, TEL.: (065) 26-511, 26-522, 23-411, FAKS: (065) 26-566

MEŠOVITO PREDUZEĆE
21000 Novi Sad
Todora Jovanovića 14

Tel: 021 396-833
Fax: 021 396-424
Tlx: 64-380

KOMPJUTERI I KOMPJUTERSKI NAMEŠTAJ



ISPORUKA ODMAH

Cene: PC XT od 21.350din, PC AT 286 od 29.950din, 386/SX od 46.150din, 386 od 62.650din, 386/33C od 111.950din, 486 od 179.050din. Povoljne cene LJ printera, plotera, OEM komponenti.

NOVO: PC TELEX KARTICE od 54.300din.

Kompjuterski nameštaj od 2.800 do 6.300din.

Printer stolovi od 5.950 do 6.300din.

Finansijki informacioni sistem 50.000din, Skladišno poslovanje 50.000din, Kompleti 35.000din, Uređenje knjigovodstva knjigovodstvenom evidencijom 40.000din, Posebni finansijski izveštaji 35.000din, Stambena zadruga 35.000din.

ZA NAJAM: 12,7% MESEČNO. SERVIS OBEZBEĐEN

MALI OGLASI

Običan mali oglas do 10 reči stoji 800 dinara a svaka dalja reč 80 dinara. Centimetar uokvirenog oglasa stoji 1 000 dinara s tim što se mogu zakupiti najmanje 2 cm i što oglas ne može biti viši od 5 cm. Postupak za objavljivanje vašeg malog oglasa je jednostavan pošaljite redakciji tekst oglasa zajedno sa foto-kopijom uplatnice kojom ste na žiro-račun RO BIGZ 60802 603-23264 (za Mala oglasa u „Računarima") uplatili odgovarajući iznos.
Rok za slanje do prvog u mesecu

YU—fontovi za Corel Draw 2.0 (Miroslavljeva, ćirilica) 011/347-967

SPAS ZA HARD DISKOVE I PODATKE
Otkazao Vam je hard-disk?! Neispravna disketa?! Ostali ste bez važnih podataka? U 99% slučajeva spasavamo disk i podatke DATA SAVE SERVICE
Tel. 021/27-425 i 021/28-285

Obnavljanje, regeneracija, ribbon pisaćih traka za sva štampača blagajne kase. Ing. I. Ešić. Fra Grge Marg... 40-940

ESCOM	Diskete
5.25" 2S/DD	1.2 DM
5.25" 2S/HD	2 DM
BASF	
5.25" 2S/DD	2 DM
5.25" 2S/HD	3 DM

Plaćanje u dinarima po tranzitnom kursu!!! Na veće količine popust!!!
Tel. 034/63-011

IBM PC & COMPATIBLE
Izrada i ponuda programa za PC računare iz svih oblasti. NOVO — BESPLATNO: Najveći izbor PD software-a (Public Dommian) u zemlji! KATALOG. Ako imate bilo kakvih problema ili nedoumica u vezi PC računara, obratite nam se sa punim poverenjem. Tradicija duga 8 godina. Besplatni savjeti inženjera, specijalista! SE software. F.G. Martića 31, 78000 Banja Luka, tel. 078/40-940

ASTROSOFT
Programi za izradu horoskopa na XT/AT računarima.
Tel. 011/167-606

Nudimo BATERIJE Nikel-Kadmijum (Ni-Cd) i Litijum (LiSOC12) za razne upotrebe i sa raznim izvodima. P.P. „Computer Electronic" Ul. Novobrdska 22, 38250 Gnjilane. Tel. 0280-28-195

DISKETE
100% ERROR FREE
5,25" 2S/DD (360 Kb) .. 21 din KOM
5,25" 2S/HD (1,2 Mb) .. 33 din KOM
3,5" 2S/DD (720 Kb) .. 27 din KOM
3,5" 2S/HD (1,44 Mb) .. 46 din KOM.

NA VEĆE KOLIČINE POPUST
NA LAGERU I BASF i NASHUA
Tel (061) 267-632

BRZA DOSTAVA!

INTERSOFT
Andrićev venac 2/1

Računari
PC AT 286/386/486
Telefax
Panasonik KX-F90
Matrični štampači
EPSON (A4,A3)

TEL.
011/340 408
011/339 104

M I K R O S A M
PREDUZEĆE ZA PROIZVODSTVO, PROMET I USLUGE, 97500 PRIJEPOLJE, SR SRB
TOPE TOPESIM S. 97500 PRIJEPOLJE YUGOSLAVIA TELEFON 099/34-972/30-007 TELEFAX 099/34-972

БУДИТЕ КОРАК ИСПРЕД!!!
ПОГЛЕДАЈТЕ САМО ДЕО НАШИХ УСЛУГА:
ЛИЧНИ РАЧУНАРИ СВАКА КОНФИГУРАЦИЈА
ШТАМПАЧИ СВИ ТИПОВИ И ФОРМАТИ.
ЛАСЕРСКИ ШТАМПАЧИ
ПОВЕЗИВАЊЕ У ЛОКАЛНУ МРЕЖУ
ОПТИЧКО АРХИВИРАЊЕ ДОКУМЕНТАЦИЈЕ.
РЕПРОМАТЕРИЈАЛИ ДИСКЕТЕ ТРАКЕ ЗА ШТАМПАЧЕ ХАРТИЈА
АПАРАТ ЗА ОБНАВЉАЊЕ ТРАКА ЗА ШТАМПАЧЕ.
ПРОГРАМИ ЗА СВЕ ВРСТЕ ПОСЛОВНИХ АКТИВНОСТИ.
ПОСЕБНА ПОНУДА ЗА ПРИВАТНА ПРЕДУЗЕЋА
КОМПЛЕТ ОПРЕМА И ПРОГРАМИ.
У3 МИКРОСАМ ВИ СТЕ ОСУЂЕИИ НА УСПЕХ !!!



Mikroračunerske mreže

NOVELL

Školovenje

INFOTRADE IZOBRAŽEVALNI CENTER u Kopru, Vojkovo nabrežje 30a organizira sledeće tečajeve za Novellove mikroračunarske mreže ze razdoblje od septembra do oktobra 1991

TEČAJ	TRAJANJE DAN	SEPTEMBAR	OKTOBAR
1 Pregled karakteristika i mogucnosti NetWare operativnih sistma 286 i 386	1	2,23	1
2. Uvod u mikroračunarske mreže	1	3 9 39	7
3 286 — Upravljač mikroračunarskih mreža	3	4	2
4 386 — Upravljač mikroračunarskih mreža	3	10	8
5 Novell — prnntanje	1	13	11
6 Instalacija NetWare 286 — workshop	2	16	14
7 instalacija Netware 386 — workshop	2	18	21
8 Novell — tehnička podrška — workshop	3	24	23

Vaše prjave i sve dodatne informacije o tecajevima možete dobiti na adresi

INFOTRADE KOPER — JAKA FLATIŠE 13 — TELEFON 064/329-523
PE KRANJ — 64000 KRANJ — TELEFAX 064/323 582

MRAK

Sonnwendgasse 32
9020 Celovec - Klagenfurt
po Rosentalerstr. mimo KGM proti centru grada, treča ulica desno.
Tel.: (9943) 463 / 35 110
Fax: (9943) 463 / 35 114

Radno vrame.
utorak, sreda, četvrtak, petak od 10 do 13. in od 15. do 18. ure
subota od 8 do 13. ure
nedelja i ponedeljak zatvoreno

DISKETE	NETO CENE
5,25" 2D	0,50 DEM
5,25" 2D HD 1,2 MB	0,86 DEM
3,5" 2DD 720 KB	0 75 DEM
3,5" 2DD HD 1 44 MB	1,50 DEM
5 25" 2D NASHUA	1,00 DEM
5 25" 2D HD NASHUA	1,20 DEM
3 5" 2D NASHUA	1,50 DEM
3 5" 2D HD NASHUA	3,20 DEM

kod kupovine većih količina popust

ŠTAMPAČI matrični, laserski, ink
NEC - STAR - CITIZEN - CANON

TVRDI DISKOVI
SEAGATE - NEC- CONNER - SYQUEST
najfiotije na koroškem

MONITORI mono, EGA, VGA
NEC - CONCORD - TARGA

MIŠEVI I SCANERI
GENIUS - UNITRON - LOGITECH

ZA UGRADNJU I SESTAVU KOMPJUTERSKIH DJELOVA KOD NAŠIH ZASTUPNIKA VAM PRIZNAMO 50% POPUSTA

Ljubljana. *RAM G, Pod gozdom 10* *tel. (061) 327 770*
ARNE *tel. (061) 559 397*
Nova Gorica: *ABAKUS d.o.o., Grčna 64* *tel. (065)21-549*

Otvorili smo servis i trgovinu v Ljubljani MRAK d.o.o. Viška 4, 61111, tel.: 061/267-748, koji je otvoren od 17. do 19. sati.
Za informacije i cjenovnik lako pozovete na tel.: 061/264-110

ELHARD

V. VLAHOVIĆA 46 25230 KULA
Tel 025-722-740
Fax. 025-723-433

XT 12 MKz ..... DK 950 DEM
AT 286-13 MKz ..... OH 1.1K1 DEM
AT 386 K KONFIGURACIJI PO VAŠOJ ŽELJI

PREPORUČUJEMO VAM AT 286-16 MHz OWS

- **KUĆIŠTE AT BABY SA ISPRAVLJAČEM**
- **MBD 286 AT-16 MKz (21 MHz LANDMARK)**
- **RAM 1 MB**
- **FLOPI MITSKKISHI 1.2 MK 5.25**
- **FDD/HDD KONTROLEK IDE 1:1**
- **GRAFIČKA KARTICA KERCULES**
- **TASTATURA 101 KLICK**
- **MONITOR MONOHROMATSKI**
- **HARD DISH WEST. IIIG. 40MB/28ms**

CENA: 1.720 DEM

IMAMO VELIKI IZBOR DODATNE OPREME ZA VAŠE LJUBIMCE I TO:

- **KUTIJA ZA DISKETE 40 KOM 5.25" ..... 20.00 DEM**
- **KUTIJA ZA KISKETE 100 KOM. 5.25" ..... 25.00 DEM**
- **MIS WITTY C-400 ..... 41.00 DEM**
- **MIS AGILEK ..... 85.50 DEM**
- **SKENER A400 (KARTICA I SOFTVEK) ..... 377.90 DEM**
- **SKEKEK MS-7000 ..... 307.70 DEM**
- **FILTER ZA MONITOR 14" KYLOK ..... 33.00 DEM**
- **FILTEK ZA MONITOR 14" ANTISTATIK ..... 53.00 DEM**
- **DISKETE 5.25" I 3.5" PO KDM ..... OD 1.00 DEM**
- **ČISTAČ GLAVE ZA FLOPI 5.25" ..... 4.40 DEM**

RAČUNARE PRODAJEMO I U DELOVIMA. GARANCIJA 12 MESECI.
VRŠIMO SERVISIRANJE I SVIH TIPOVA RAČUNARA.
POZOVITE NAS OD 6h DO 18h SVAKOG DANA. SUBOTOM OD 6h DO 12h.
CENE SU ZA KRAJNJEG KUPCA.

COMPUTER DIVISION

NAJVEĆI IZBOR LICENČNOG SW.
DOS, MAC, OS/2, VAX,
Unix-Interactive, Unix-SCO

NAJVEĆI CD -ROM KATALOG

Informacioni klub BIGAI
* 20 000 HW produkata
* 30 000 SW produkata
* 10 000 COMM produkata
* 60 000 članaka iz područja informatike
* Zahtevajte informacije o članstvu
tel: 061 310 660

PONUDA MESECA
Mitsubishi Diamond Scan
14" monitor
1280 x 1024 noninterlaced

NOVO - NOVO
INGRESS ,FOCUS
UNIX DOS IBM VAX

TRIAS
WTC
Dalmatinova 4 61000 Ljubljana
tel: (061) 316 343, 110 033, 310 660

YUGO SOFTWARE
* Financijsko poslovanje
* Trgovinsko poslovanje
* Gostinsko poslovanje
* Stomatološki Informacioni Sistem COMPUDENT
* Automatizovana Kancelarija CASSIOPEA
* ATM+ YU FONTS
* Komunikacijski direktor OPERATOR
* LECTURUS OCR
* VEKTOR vektorizacija
* ALADIN poslovni sekretar
tel: 061 110 033

COREL SYSTEM & TRIAS WTC
CORELDraw! 2.0
YU fonti, PAL kasete

RANK XEROX & TRIAS WTC
YU - VENTURA

AUTHORIZED TRAINING
VENTURA PUBLISHER
CorelDRAW!
tel: 061 316-343

PONUDA MESECA
MS DOS 5.0 UPGRADE

MOGUČNOST PLAČANJA U DEVIZAMA - jeftinije do 70% - POZOVITE !

SHORT FORM CATALOG 1991

IEEE-488 and VXIbus Control, Data Acquisition, and Analysis

...the Most Choices



MULTIFUNKCIJSKE I/0 KARTICE (AD DA DIGITAL I/0, COUNTER, TIMER)

IEEE 488 DODACI ZA SVE VRSTE RAČUNARA

MODULI ZA KONDICIONIRANJE SIGNALA

DATA ACQUISITION SOFTWARE (LABWINDOWS LAB VIEW ASYST LABTECH NOTEBOOK, LABTECH, MEASURE ltd

VXI INSTRUMENTACIJA

GANTNER INTELIGENTNI SENZOR MODULI

HIGH OUALITY RECORDERI (ABB GOERTZ, GRAPHTEC)

ALR — ADVANCED LOGIC RESEARCH PERSONALNI RAČUNRI

APLIKACIJE NA KLJUČ

OEWETRON

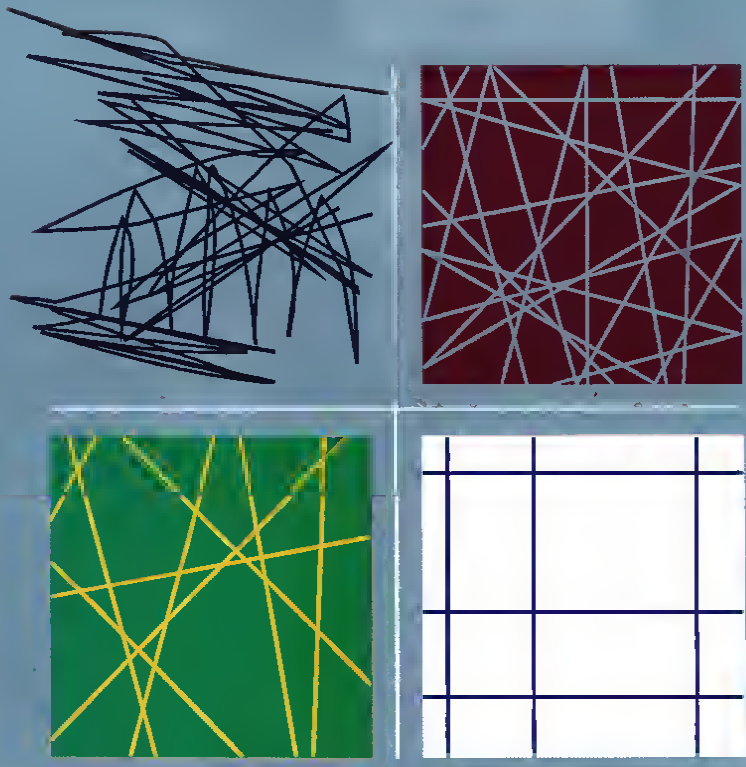
DEWETRON d.o.o.

Krakovo 1 YU 61433 RADEČE Tel./Fax 0601/81 935

Zastupstvo NATIONAL INSTRUMENTS, GANTNER MEILHAUS, GRAPHTEC



TRIESTE (Italy) - Via Caboto 19 - Tel. (040) 82.34.21 - Fax (040) 82.34.25



The Art of Networking

Mreže su novi korak u razvoju računarstva i informatike. Vreme individualnih i velikih mainframe računara u poslovnom okruženju je prošlo. Računara treba povezati u mreže, da bi se optimalno iskoristili ljudski i mašinski resursi.

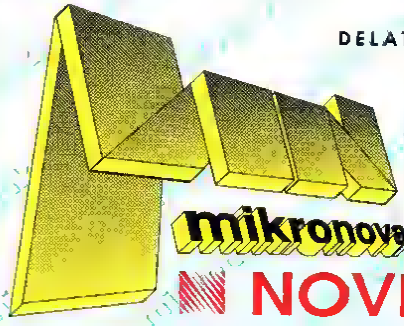
Da bi se formirala mreža nije dovoljno kupiti kutije. Naši stručnjaci su kompetentni konsultanti koji mogu rešiti Vaše probleme u oblasti poslovne obrade. Ako kod nas dođete samo sa znanjem o potrebama koja sistem treba da ispuni, izradićemo projekat kompletnog informacionog sistema. Poslovni softver koji vam nudimo predstavlja vrhunske dostignuća u ovoj oblasti.

Odabrali smo "Novell", najvećeg proizvođača softvera za lokalne računarske mreže, jer se "Most" pouzdaje samo u proizvođače koji garantuju kvalitet. Sa "Novell" NetWare operativnim sistemima naše mreže ispunjavaju sve Vaše zahteve za povezivanja PC, mini i mainframe računara. Raspolažemo znanjem i iskustvom koja Vam garantuje da će Vaše mreže raditi uz maksimalno iskorišćenje performansi. Sva kvarova u radu mreže otklanjamo u roku od 24 časa. Telefonska podrška uvek Vam je na raspolaganju.

Vaša kupovina u "Mostu" tako postaje rešenje po sistemu "ključ u ruke". Nudimo Vam sistema projektovane da budu tehnički fleksibilni i finansijski povoljni. Obratite nam se zahtevom za ponudu za Vaš konkretan problem, ili zahtevajte listu naših referenci.

MOST

PREDUZEĆE "MOST", TEL. 422-170, FAX 426-045, BEOGRAD, KRALJA PETRA 2



mikronova

DELATNOST : Projektovanje i izgradnja informacionih sistema

- PRODAJA NOVELOVIH PROIZVODA
- PROJEKTOVANJE informacionih sistema
- PROIZVODNJA kompletnog hardware-a (računarske opreme)
- PROJEKTOVANJE I IZGRADNJA računarskih mreža i komunikacija
- IZRADA software-a (aplikativnih programa)
- ODRŽAVANJE I SERVISIRANJE harware-a i software-a
- NABAVKA repromaterijala (diskete, ribani, perforirani papir)
- OBUKA korisnika

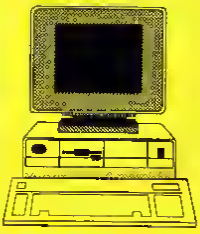
NOVELL® AUTHORISED DEALER

NOVELL®

RAČUNARSKE MREŽE

- NETWARE Ver 2.2
- NETWARE Ver 3.11
- MREŽNE KARTICE I OSTALA MREŽNA OPREMA

HARDWARE



Proizvodnja i isporuka najsavremenije računarske opreme, u konfiguracijama prema zahtevu korisnika:

- PC/AT-286 kompatibilni računari
- PC - 386 kompatibilni računari
- PC - 486 kompatibilni računari

- Hard diskovi
- Flopi diskovi
- Grafičke karte
- Monitori visoke rezolucije
- FAX karte i Modemi
- Matrični, laserski i heavy duty štampači

SOFTWARE

Proizvodnja optimalno koncipovanog aplikativnog software-a za IBM PC/XT/AT - kompatibilne računare

- OPTIMALNA UPOTREBA POLJOPRIVREDNIH MAŠINA
- MATERIJALNO I FINANSIJSKO POSLOVANJE
- KADROVSKA EVIDENCIJA, LIČNI DOHOTCI I KREDITI
- UPRAVE DRUŠTVENIH PRIHODA
- KLINIKE I BOLNIČKA POSLOVANJA,
- GINEKOLOŠKO AKUŠERSKE KLINIKE
- STAMBENA PREDUZEĆA
- UNIVERZALNI PAKETI ZA STATISTIKU, TRGOVINSKA PREDUZEĆA, ADVOKATSKE KANCELARUE, OBRAČUN KAMATA, BUDŽETSKO KNJIGOVODSTVO
- UNIVERZALNI GENERATOR IZVEŠTAJA
- PROGRAMI PO ZAHTEVU KORISNIKA

NOVO

CLIPPER USER KLUB U OSNIVANJU!
Sve najnovije dostignuce i informacije direktno od NANTUCKET-a, javite se!

PROVERITE NAŠU REFERENTNU LISTU

DILERI DOBRODOŠLI !

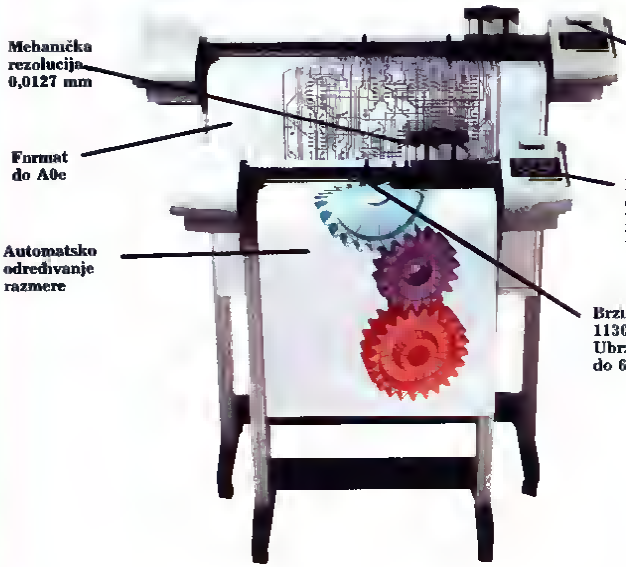


NAŠA NOVA ADRESA JE:
P.P.mikronova D.O.O. Novi Sad, Lenjinov trg 5/II
Tel.021/52-999, 21-077 Fax.021/28-042

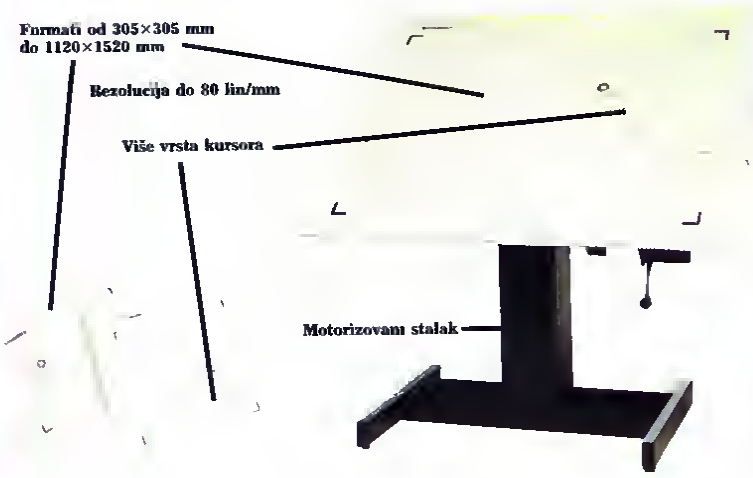
HOUSTON INSTRUMENT

Summagraphics™

Ploteri



Digitalizatori



Skeneri

Sekači

Vrhunske grafičke periferije, softver i inženjering

Grafička tabla „Summagraphics“ na poklon! SVI KUPCI PLOTERA AO i A1, SKENERA I SEKAČA DOBIJAJU GRAFIČKU TABLU „SUMMASKETCH II“ GRATIS!

Ovlašćene prodajne organizacije:

Kma D.O.O.
Milentija Popovića 9
BEOGRAD (Sava centar)
Tel.: 011/222-19-28
Tel.: 011/222-43-22, lok 95
Fax: 011/455-785

SoftWell d.o.o.
Kompjuterski inženjering
Pozorišni trg 1
21000 Novi Sad
Tel: 021/51-999 52-366

El „Računari“ – Niš
Bul. V. Vlahovića 80–84
18000 Niš
Tel.: 018/325-461 55-583
Fax 325-726

Lanaco
Braća Pavlića H
78000 Banjaluka
Tel.: 078/33-992, 48-717
Fax 33-992

Makpetrol kompjuteri
Kompjuterski inženjering
Ul. „Orce Nikolov“ bb
91000 Skopje
Tel. 091/229-976, 221-040
Fax 229-676

assist

S.I.S. QUALITY PRODUCT

za profesionalce



- zaštitni ekranski filteri
- nosači računara
- stolovi za printere
- prigušivači zvuka

■ zastupal

MEDIS

MEDIS, Titova 58 61000 Ljubljana
Telefon & telefax (061) 329 270

COMTRAD

VOLKS-AT 286/12

1.991.- DM



ISPORUKA ODMAH!

CONFIGURATION

- 80286 INTEL CPU & TI CHIPSET
- 12 MHZ SPEED
- LANDMARK 15.6
- 1 MB RAM INCLUDED
- 2 SERIAL PORTS
- 1 PARALLEL PORT
- 1 GAME I/O PORT
- SOCKET FOR 80287 COPOCESSOR
- FLOPPY DISC DRIVE 5.25" 1.2 MB
- FLOPPY DISC DRIVE 3.5 " 1.44 MB
- 40 MBYTE AT-BUS HARDDISC
- VGA DISPLAY ADAPTER
- 14" VGA COLOR MONITOR
- MF II KEYBOARD, 102 KEYS
- SERIAL MOUSE 2 BUTTOM

Build for your satisfaction

386SX (ista konfiguracija) 2.499 DM

SEIKOSHA SP 2000 Ai	SEIKOSHA SL 92A
9 iglica/192 cps	24 iglice/240 cps
DM 361	DM 595

COUPON

Please send me information about:

- 386 SX
- 386 CACHE 33
- 486 ISA
- 486 EISA 33

ComTrad
HIGH SPEED PC S

COMTRAD WORLDWIDE SUPPORT
UNITED STATES • GERMANY • VENEZUELA
YUGOSLAVIA • NETHERLANDS • SPAIN

ComTrad

Central European Office
Wehlerstrasse 12, Wehler Park
4000 Dusseldorf 30
West Germany
Tel 99-49 211-65-08-0
Fax 99-49-211-65-08-40

ComTrad Information
Vladimira Popovica 6
11000 Beograd
Tel 011-2224-151
Tel 011-2222-652
Tel 011-2224-433/8265
Fax 011-2223-951

ComTrad Zagreb
Krezmina 4
41000 Zagreb
Tel 041-211 873
Fax 041-211-873

KMA d o o
Mlientije Popoviće 9
Tel 011-2221-928
Tel 011-2224-322
lok 615
Fax 011-455-785

AUTODESK

AutoCAD R 11
AEC
AutoSHADE
AutoSKETCH
ANIMATOR
3D STUDIO

RASTEREX Graphics solutions

PROFESIONALNE
GRAFIČKE KARTICE:
LIGHT AT,LIBERTY
AutoIcon
RxImage
SpotLight

océ Graphics

PLOTERI SA PERIMA
TERMALNI PLOTERI
GRAFIČKE TABLE
DIGITAJZERI
KOLOR TERMALNI
PRINTERI

genoa

OSNOVNE PLOČE.
386-33 MHz/64k,
486-33 MHz/256k
GRAFIČKE KARTICE



Chicony

TASTATURE

LAPTOP RAČUNARI

PROJEKTOVANJE SISTEMA
PC RAČUNARI SVIH KONFIGURACIJA
NOVELL MREZE
AutoCAD I POSLOVNE APLIKACIJE

RAČUNARSKI INŽENJERING
BEOGRAD

RAČUNARSKI INŽENJERING
COMPUTER ENGINEERING

Sedište/Head office
Mice Stojkovica 8,
11130 Beograd

Predstavništvo/Representative office
Šumadijski trg 6a,
11030 Beograd,

tel 38 11 557-592
fax 38 11 559-845

Ploteri Océ Graphics
Evropa je odabrala kvalitet



Izbor naših plotera je veoma širok! (na žalost ova stranica n

Ploteri sa perima svih formata
Termalni ploteri
Grafičke table i digitajzeri
Color POSTSCRIPT termalni printeri

Najkonkurentnije cene vodećeg evropskog proizvodača
Širok izbor potrošnog materijala
Obezbeden servis



océ Graphics

Ekskluzivni distributer:

Računarski inženjering

OVLAŠĆENE PRODAJNE ORGANIZACIJE

„FONI"
11300 Smederevo
Tel/fax (026)24-544

„BEE"
1800 Niš
Tel (018)62-063
Fax (018)47-424

„COBEX"
Titograd
Tel/fax (081)24 149

„ARDI ENGINEERING"
Priština
Tel (038)22 981
Fax (38)23 578

Beograd, tel: (011) 557-592, fax (011) 559-845

Quo Vadis ?

ABV Graphics



AUSTIN

Profesionalna rešenja i profesionalna oprema

PERSONALNI RAČUNARI / RADNE STANICE: **Austin Computer Systems, CompuAdd Corporation Computers;** CAD ***Nth Graphics*** CAD kontroleri i softverski emulatori, ***Pixelworks, Inc.*** programabilni CAD kontroleri (za AutoCAD, VersaCAD, Computervision Personal Machinist, --), FULL COLOR COMPUTING: **Truevision TARGA, Vista;**, **AT&T GSL TOPAS, Howtek, Mitsubishi;** RAČUNARSKI SISTEMI OPŠTE NAMENE: **Bull HN DPX/2** UNIX računarski sistemi, **PROGRESS** 4GL relaciona baza podataka, **TenX Technologies** WORM i Erasable optički diskovi

ABV Graphics, Andrićev Venac 4, 11000 Beograd, tel/fax: 011/345-277

Distributer u Sloveniji: CGS d o o , Ljubljana, tel: 061/347 661



MOST design, foto Nenad Marjanović, skulptura Marko Krotofil

Zašto bi vaš novi kompjuter trebalo da bude "M O S T"

U "Mostu" se računari ne prodaju po svaku cenu - prodajemo samo najbolje. Kod nas ćete dobiti savet, pomoć u radu ili projekat za kompletan informacioni sistem. Kompjuteri i informatika su naš životni poziv, pa možete biti sigurni da ćete kod nas uvek dobiti kompetentnu stručnu pomoć. Možete nazvati telefonom - možda tako rešimo vaš problem.

Ciljevi naše kompanije uvek su bili jasni: stvoriti bazu zadovoljnih korisnika, dostići i održavati visok nivo kvaliteta, postati najbolji. Mislimo da su ti ciljevi ostvareni, a vi ste tome doprineli. Danas biti najbolji određuje koje će kompanije preživeti. "Most" će još dugo biti sa vama.

Mi se razlikujemo od drugih. Trudimo da sjedinimo umetnost i tehnologiju. Naš oglas više nije običan cenovnik.

JUNIOR: 286/16 • ROSE: 386SX/16, 386SX/20 • ASSISTANT: 386/33 • GENIUS: 386/40 • DOLPHIN: 486SX/20 • STAR: 486/25 • GOLIATH: 486/33

NOVO: NetWare-Ready diskovi - tvrdi diskovi spremni za instalaciju "Novell" NetWare mreža, sa obavljenom analizom površine diska.

PREDUZEĆE "MOST" TEL. 422 170 FAX 426 144 BEOGRAD KRALJA BODINA 2

EPSON

VELIKO IME, POUZDAN KVALITET!

Izključno naše izvode izmed ostalih prodajajo:

Avtotehna, Titova 36 • Celovška 228 Ljubljana
Mladinska knjiga Veletrgovina, Titova 145 Ljubljana
Mladinska knjiga Trgovina Titova 3 Ljubljana
Gambit p o , Titova 8 Ljubljana
ATR Ljubljana Birostroj

Mogucnost leasing prodaje!



d.o.o.
CELOVŠKA 175 • YU
61107 LJUBLJANA
REPRO LJUBLJANA
TELEFON 061/552-341 552-150 554-450 FAX (061) 552-563
TLX 31 639 yu-antena p p 69

POUZDANO IME, VELIKI KVALITET!



Industrijski računari i za najteže poslove

Osobine računara
TEXAS MICRO SYSTEMS

- visok stepen pouzdanosti sistema (MTBF ugrađenih komponenata 70—100 hiljada sati)
- otpornost na udare (do 25 g)
- otpornost na vibracije ()do 1,5 g)
- prilagođenost za rad u posebnim uslovima (visoka relativna vlažnost, visoka temperatura, prašina)
- 80286, 80386 i 80486 CPU

• Svaki računar je konfigurisan u tvornici i ispitan (burn in test na 55° C)

GENERALNI DISTRIBUTER ZA JUGOSLAVIJU.

METRONIK

Metronik d.o.o., Ljubljana

Ptujska 19
61113 Ljubljana
telefon 061/182-533
telefax 061/349-290

Pored toga, nudimo vam još:

- sisteme za akviziciju podataka i programibilne sisteme upravljanja
  opto 22 — SAD
- SCADA programske pakete — THE FIX
  Intellution — SAD

Posetite nas na sajmu „SODOBNA ELEKTRONIKA 91“ u Ljubljani 7—11. 10. 1991, u hali B/I, prostor br. 20

SCO
THE SANTA CRUZ OPERATION

Softverska kompanija 1990. godine

Operativna okruženja

SCO System V

Porodica SCO Sistema V obuhvata SCO UNIX System V/386 Release 3.2 i SCo XENIX System V, višekorisnički, multitasking operativni sistemi za mikroračunare bazirani na industrijskim standardima

SCO UIX System V/386 Release 3.2

Robustna, potpuno licencna verzija AT&T UNIX Sistema V, sa dodatnim poboljšanjima koja je isključivo ugradio SCO

SCO XENIX

Industrijski standardan UNIX sistem za personalne računare bazirane na Intelu

SCO MPX

Multiprocesorsko proširenje za računare bazirane na 386 i 486 koji rade pod SCO UNIX Systemom V/386

SCO VPfix

Pod SCO VPfix, višestruke MS DOS aplikacije mogu tel paralelno sa konvencionalnim XENIX i UNIX sistemima

Komunikacije i mreže

SCO TCP/IP

Povezivanje preko mreže između razmih operativnih sistema i računara

SCO NFS

Networked File i Data Sharing za SCO UNIX i Open Desktop

SCO XENIX-NET

Microsoft mreže za XENIX sisteme. Lokalna mreža namenjena da specifično poveže personalne računare koji rade pod XENIX-om i DOS-om

SCO UniPATH SNA-3270

Višekorisnička veza između mikro i glavnog računara — ulaz višekorisničkih personalaca u IBM SNA mreže

Radne stanice

Open Deskop

Potpun grafički operativni sistem. Open Desktop vam omogućava da koristite moćne grafičke aplikacije u prozorima, da manipulišete podacima, pristupate datotekama i aplikacijama na umreženim sistemima i da koristite DOS i UNIX sisteme

Poslovna automatizacija

Microsoft Word 5.0

Višekorisnički tekst procesor za UNIX sisteme

SCO Portfolio

User-friendly interfejs za SCO System V uključujući aplikacije za automatizaciju kancelarijskog poslovanja kao što su elektronska pošta (e-mail) i kalendar

SCO Professional

Lotus 1-2-3 workalike za SCO System V

SCO FoxBASE+

dBASE III plus workalike za SCO System V

Svetski lider u rešenjima za softver otvorenih sistema

MAC RITA AWARDS PC MAGAZINE COMPUTING

SCO je dobio priznanje kao Softverska Kompanija Godine na ceremoniji dodele RITA nagrada



DISTRIBUTER ZA JUGOSLAVIJU

ENERGOPROJEKT
HOLDING KORPORACIJA

MDD ENERGODATA

11070 Beograd, Lenjinov bulevar 12, p. fah 20
Tel (011) 138-520, 139-712, 222-3072
Fah (011) 222-4780 • Tlx ENDATA YU 11764

WordPerfect® familija

WordPerfect

vodeći tekst procesor na svjetskom tržištu
preko 4,000,000 korisnika
jedinstvena sprega snage i jednostavnosti
preko 500 podržanih printera
tabele, matematika, jednadžbe
prethodni pregled dokumenta
integracija grafike u tekst

DataPerfect

relacijska baza podataka
jednostavno generiranje ekrana
jednostavno generiranje izvještaja
fleksibilni pregled podataka
run-time modul

WordPerfect OFFICE

integracija svih WP produkata
potpuno rješenje automatizacije ureda
pristup do drugih aplikacija
elektronička pošta (e-mail)
grupni terminski kalendar
WAN i globalna komunikacija
kalkulator, rokovnik



domaća verzija!!!

PlanPerfect

moćan tabelarni kalkulator
izvanredne grafičke mogućnosti
optimizacija preračunavanja
preko 2,000,000 ćelija po tabeli
preko 100 specijaliziranih funkcija
upotreba virtuelne memorije
povezivanje tabela

DrawPerfect

vrhunski paket za poslovnu grafiku
preko 500 predefiniranih slika
tabele, grafikon
vektorska definicija objekata
ulaz podataka iz 14 grafičkih formata
izlaz na jedinice visoke rezolucije
30 matičnih fontova
run-time modul

SAMO NAJBOLJE JE DOVOLJNO DOBRO!

perpetuum doo.
integralni informacijski sistemi
Kuševina 5 41000 Zagreb, tel 317-020; 328-999, fax.328-999

MCH Computer d.o.o.
62000 Maribor Tomsiceva 19
Tel (062) 28 250 26 091
Fax (062) 28 250

MCH Solution d.o.o.
11070 N Beograd Omladinskih brigad 104
Tel (011) 154 904
Fax (011) 161 445

MCH Technologies d.o.o.
41000 Zagreb Proleterskih brigada 78
Tel (041) 539 892
Fax (041) 538 446

UZMITE KVALITET, TO STE ZASLUŽILI !

Zastupamo svetski poznata imena

MCH Computer Systeme
PEACOCK
MITSUBISHI
SIGMA DESINGNS
TALLGRASS
WESTERN DIGITAL

Nudimo vama

Računarsku opremu
Software
Mreže
Servis
Komponente
i svu ostalu podršku

COMPUTER SHOP

MCH Computer-Systeme Ges.m.b.H. A - 8472 Strasss / Stmk., Hofgreith 2, Tel. 00 34 53 44 50, Fax. 00 34 53 43 65

PONUDA RAČUNARSKE OPREME

Racunalnik	40 MB	85 MB	112 MB	212 MB
AT 286-12 MHz	1 299.-	1 614.-	1 899.-	2 229.-
AT 286-16 MHz	1 345.-	1 660.-	1 945.-	2 275.-
AT 386SX-16 MHz	1 760.-	2 095.-	2 380.-	2 710.-
AT 386-25 MHz	2 146.-	2 461.-	2 746.-	3 076.-
AT 386-25C MHz, 32 KB Cache	2 235.-	2 545.-	2 830.-	3 160.-
AT 386-33C MHz, 64 KB Cache	2 466.-	2 776.-	3 066.-	3 396.-

OPCIJE	doplata
mono VGA 640 x 480	172.-
color VGA 1024 x 768	634.-
dodatni 1 MB RAM	91.-
dodatni FDD	130.-
mini tower	20.-

Sve cene su u DEM (Neto)

Uz svaku konfiguraciju ide
- 1 MB RAM kućište AT desktop FDD (1.2MB ili 1.44 MB)
- I/O kartica hercules 14 monokromatski monitor
- tastatura CHERRY YU

Garancija 1 godina u MARIBORU
Servisi u Mariboru Zagrebu i Beogradu

VRLO POVOLJNO : MCH 286-12 MHz, 1MB RAM 14" color monitor (1024 x 768) I/O kartica 42 MB tvrdi disk tastatura CHERRY YU graf. kartica VGA (1024 x 768) 1.2 MB floppy disk

CENA : 1 933,- DEM

MCH computer systeme

AUTRONIC

MCH Computer-Systeme
Handelsgesellschaft m.b.H
8472 Stras/Stmk. Hofgreith 2
Tel.: 9943 34 53 44 50
Fax.: 9943 34 53 43 65

PEACOCK COMPUTER

AUTRONIC Computer Systeme
A 9020 Klagenfurt, Radetzkystr. 18
Tel.: 9943 463 51 48 71
Fax.: 9943 463 51 48 73

Osnovne konfiguracije

AT 286 - 16	1 MB RAM, VGA, 40 MB tvrdi disk, DOS 4.01	47.821,- DIN
AT 286 - 16	1 MB RAM, VGA, 110 MB tvrdi disk, DOS 4.01	63.279,- DIN
AT 386 - 25	2 MB RAM, VGA, 40 MB tvrdi disk, DOS 4.01, WINDOWS 3.0 + MS Mouse	81.282,- DIN
AT 386 - 25	2 MB RAM, VGA, 110 MB tvrdi disk, DOS 4.01, WINDOWS 3.0 + MS Mouse	96.740,- DIN
AT 386 - 25C	4 MB RAM, VGA, 110 MB tvrdi disk, DOS 4.01, WINDOWS 3.0 + MS Mouse	114.153,- DIN
AT 386 - 33C	4 MB RAM, VGA, 130 MB tvrdi disk, DOS 4.01, WINDOWS 3.0 + MS Mouse	127.159,- DIN
AT 486 - 25C	4 MB RAM, VGA, 110 MB tvrdi disk, DOS 4.01, WINDOWS 3.0 + MS Mouse	163.259,- DIN
MONITOR	VGA MONO MONITOR 640 x480	5.941,- DIN
MONITOR	VGA TRISCAN MONITOR 1024 x 768	22.843,- DIN
	NOTEBOOK 80386 SX 136.188,- DINARA	

Distributeri

MCH Computer d.o.o.
62000 Maribor, Tomšičeva 19, Tel.: & Fax: (062) 28 250

MCH Solution d.o.o.
11000 Beograd, Zaplanjska 86, Tel.: (011) 468 732,
Fax: (011) 467 059

MCH Tehnologies d.o.o.
41000 Zagreb, Proleterskih brigad 78, Tel.: (041) 539 892,
Fax: (041) 538 946

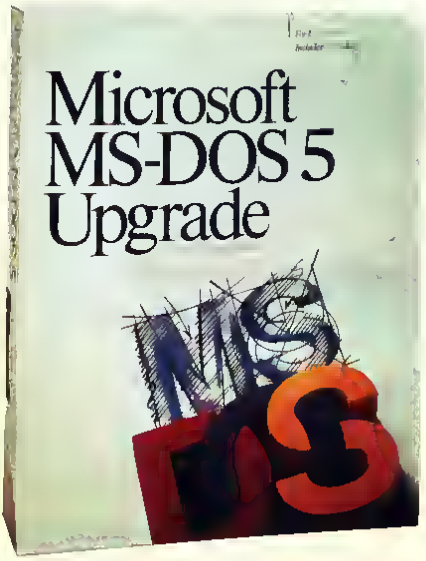
AUTRONIC d.o.o.
61000 Ljubljana, Kardeljeva ploščad 17
Tel.: (061) 345 161
Tel.: & Fax: (061) 302 581

AUTRONIC d.o.o.
41000 Zagreb, Kollerova 3
Tel.: & Fax: (041) 232 259



PRELAZNI ROK

Pređite još danas sa nepotpune i zastarele piratske kopije na novi Microsoft DOS 5.0 pomoću ...



Microsoft DOS 5.0 Upgrade je na raspolaganju svim korisnicima, koji već imaju MS-DOS na PC računaru. Takođe onima koji su dosada radili na piratskoj kopiji.

Potpuno nova verzija Microsoftovog operacionog sistema koji inače pokreće 60 miliona PC računara širom sveta sa sobom donosi:

YU slova – napokon su naša slova originalno podržana od strane operacionog sistema sa codepage 852
Task Switcher – prelaz iz jedne u drugu aplikaciju jednim pritiskom na tastaturu, više memorije (do 620 K) na raspolaganju za vaše aplikacije
DOS Shell – rad sa fajlovima preko jednostavnog i preglednog user interfejsa
QBasic – moderan i strukturiran Basic programski jezik
Editor – ekranski editor teksta

Prodaja Microsoft DOS 5 Upgrade i ostale programske opreme Microsoft sa registracijom u Jugoslaviji

CET Beograd, tel. 011/336-630
Jugodata Beograd, tel. 011/453-560
REY Beograd, tel 011/488-5472
Softwell Novi Sad, tel: 021/51-999
Mladinska knjiga Zagreb, tel: 041/422-460
INTES zagreb, tel: 416-162
MicroLab Zagreb, tel 041/692-703
Almido Split, tel: 058/43-137
Print Zagreb,tel. 041/412-843
Medija Ljubljana, 061/221-838
7 L Murska Sobota, tel 069/21-353
Mikrohit Ljubljana, tel: 215-042
CAT Ljubljana, tel: 061/223-949
SPIN Nova Gorica, tel 065/26-800
Monitoring Zagorje, tel: 0601/81-935

Za sve kupce do 15. oktobra nagradno izvlačenje Microsoftovih proizvoda na standu Atlantis na sajmu INTERBIRO u Zagrebu

Distribucija Microsoft programske opreme u Jugoslaviji

ATLANTIS d.o.o.
POSLOVNI INFORMACIJSKI SISTEMI
Cankarjeva 4
61000 Ljubljana
tel/fax
(061) 221 608

ATLANTIS

Microsoft je zaštitni znak Microsoft, Inc